

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Уральский государственный педагогический университет»
Институт математики, физики, информатики и технологий
Кафедра информатики, информационных технологий и методики обучения
информатике

ИНТЕРАКТИВНАЯ КАРТА ДЛЯ ПОСТРОЕНИЯ ТУРИСТИЧЕСКИХ МАРШРУТОВ ПО ДОСТОПРИМЕЧАТЕЛЬНОСТЯМ УРАЛА

Выпускная квалификационная работа

Квалификационная работа
допущена к защите
Зав. кафедрой

дата

подпись

Исполнитель:
студент 5 курса
заочной формы обучения
профиль ПИвС 1501z
Меньщиков Игорь Борисович

подпись

Руководитель:
старший преподаватель,
Старкова Людмила Николаевна

подпись

Екатеринбург-2020

Содержание:

ВВЕДЕНИЕ.....	3
ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ СОЗДАНИЯ ИНТЕРАКТИВНОЙ КАРТЫ	4
1.1 Обзор современных интерактивных технологий	4
1.2 Анализ веб-инструментов для создания интерактивных карт	12
1.3 Техническое задание на разработку интерактивной карты.....	19
ГЛАВА 2. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНТЕРАКТИВНОЙ КАРТЫ ДЛЯ ПОСТРОЕНИЯ ТУРИСТИЧЕСКИХ МАРШРУТОВ ПО ДОСТОПРИМЕЧАТЕЛЬНОСТЯМ УРАЛА	22
2.1 Проектирование интерактивной карты достопримечательностей Урала.....	22
2.2 Описание процесса разработки интерактивной карты	23
2.3 Апробация внедрения	48
ЗАКЛЮЧЕНИЕ	51
СПИСОК ИСТОЧНИКОВ	52
ПРИЛОЖЕНИЯ.....	56

ВВЕДЕНИЕ

В современном мире интерактивность все глубже проникает в жизнь людей. Интерактивность – это важный и очень действенный маркетинговый инструмент, обеспечивающий увеличение посещаемости сайта, а также увеличение вовлеченности пользователя.

Интерактивность позволяет существенно повысить эффективность взаимодействия участников коммуникации. На практике интерактивные элементы взаимодействуют с пользователем, адаптируют предложение под его потребности. Качественный интерактив помогает выделиться среди конкурентов. При выборе дорогостоящих товаров, предназначенных для длительной эксплуатации, пользователь может перелопатить целый каскад веб-страниц. Удачный интерактив помогает лучше запомниться такому потенциальному покупателю, расположить его к компании [6].

Целью данной работы является разработка интерактивной карты для построения туристических маршрутов по достопримечательностям Урала.

Для достижения поставленной цели необходимо решить следующие задачи:

1. рассмотреть и проанализировать современные интерактивные технологии;
2. выбрать веб-инструмент для создания интерактивной карты;
3. спроектировать и разработать интерактивную карту для построения туристических маршрутов по достопримечательностям Урала.

Решение определённых в работе задач будет осуществляться на базе портала «Наш Урал» (<https://nashural.ru>). Разработанная интерактивная карта будет использоваться посетителями портала.

ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ СОЗДАНИЯ ИНТЕРАКТИВНОЙ КАРТЫ

1.1 Обзор современных интерактивных технологий

JavaScript — это язык программирования, который широко применяют для придания интерактивности веб-страницам. Когда на веб-странице происходит какой-то интерактив — отображение постоянно обновляемого содержимого, или интерактивных карт, или анимация графики, или воспроизведение видео и т.д. — можно быть уверенным, что скорее всего, это JavaScript [5].

JavaScript был создан Бренданом Айком, работавшим на Netscape в 1995 году. Первая версия языка работала только в браузере Netscape и предлагала узкую функциональность, но благодаря сообществу разработчиков он продолжил развиваться и теперь до 92% сайтов используют его. В ноябре 1996 года JS был стандартизирован [14]. Этот стандарт получил имя ECMAScript. Современный стандарт ECMAScript 5 включает в себя новые возможности для разработки, ECMAScript 6 будет шагом вперёд в улучшении синтаксиса языка.

JavaScript является интерпретируемым языком — код запускается сверху вниз и результат запуска немедленно возвращается. Нет необходимости преобразовывать код в другую форму, перед запуском в браузере. Помогает в этом специальная программа, которая называется «движок» JavaScript или интерпретатор JavaScript. Он выполняет три функции:

1. Движок читает («парсит») код скрипта;
2. Затем он преобразует («компилирует») скрипт в машинный язык.
3. После этого машинный код запускается и работает достаточно быстро.

Кроме того, интерпретатор JavaScript проводит оптимизацию на каждом этапе. Именно поэтому программы, написанные на JS, работают настолько быстро, что пользователи не замечают никаких дополнительных подзагрузок.

В настоящее время во все браузеры встроены различные движки JavaScript (например, V8 – в Chrome или SpiderMonkey – в Firefox), именно поэтому они могут выполнять скрипты на странице. Но это не означает, что JavaScript используется исключительно в браузере. Программы, написанные на этом языке, можно запускать на любом веб-сервере или другом устройстве, если предварительно там будет установлен интерпретатор [10].

JavaScript – быстрый и мощный язык, но браузер накладывает на его исполнение некоторые ограничения. Это сделано для безопасности пользователей, чтобы злоумышленник не мог с помощью JavaScript получить личные данные или как-то навредить компьютеру пользователя. Так, например, программы на JavaScript не имеют никакого низкоуровневого доступа для работы с памятью и процессором [33].

Но это не означает, что JS всего этого не умеет: все зависит от окружения, в котором запущен JavaScript. Например, с помощью плагинов и расширений для браузеров, можно расширить возможности JS в браузере, но для этого пользователь должен самостоятельно установить такое расширение. На сервере и вовсе никаких ограничений для JS нет [26].

В браузере JS обладает возможностью делать то, что связано с взаимодействием с пользователем, манипуляциями со веб-страницей, а также с сервером:

- Динамическое изменение содержимого веб-страницы. Создание, удаление и редактирование HTML-кода, модификация стилей элементов веб-страницы;
- Реагирование на различные действия посетителя веб-страницы. JS умеет обрабатывать клики мыши, нажатия на клавиши клавиатуры, понимает перемещения курсора мыши;
- JS посылает запросы на веб-сервер и позволяет загружать различные данные без перезагрузки текущей веб-страницы, благодаря такой технологии, как AJAX;

- Полная работа с cookie пользователя, а также взаимодействие с самим пользователем, средствами вывода сообщений или заполнения различных форм;
- Умеет хранить данные на стороне пользователя.

Однако, большинство функций JS в браузере ограничено лишь текущим окном и веб-страницей.

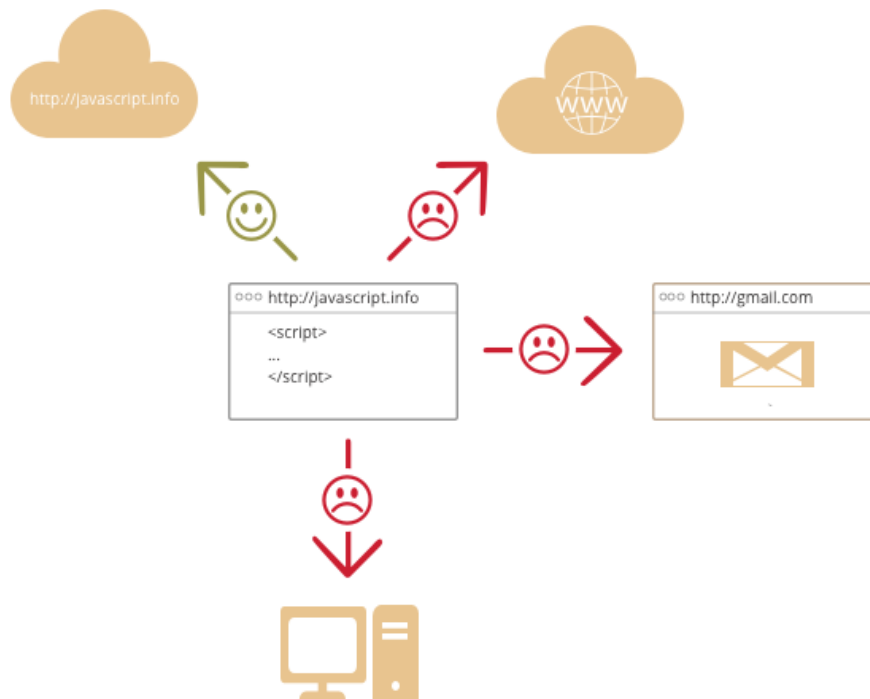


Рис 1 – Возможность доступа к данным с помощью JavaScript

- JS не может самостоятельно читать, записывать или копировать разнообразные данные на жесткий диск (стоит отметить, что браузеры могут предоставить JS работу с файлами, но эта возможность ограничена специально выделенной директорией) или запускать другие программы. JS не имеет прямого доступа к операционной системе, хотя возможности по доступу к различным устройствам уже присутствуют в современных стандартах и доступны в некоторых браузерах [31].
- JS не может обращаться к другим вкладкам и окнам браузера, только если он сам открыл эти окна или вкладки из одного источника (например, одинаковый домен). Стоит упомянуть, что существуют способы обхода этого ограничения с помощью специального кода на каждой странице.

- JavaScript может посылать запросы на сервер, с которого пришла страница. Запрос на другой домен возможен, но и здесь есть ограничения безопасности.

Отдельно необходимо отметить еще несколько особенностей JS:

- Полная интеграция с HTML и CSS.
- Простые вещи делаются просто.
- Поддерживается всеми распространёнными браузерами и включён по умолчанию, нет необходимости переживать, что функционал будет недоступен у части пользователей.

Эти три особенности одновременно присутствуют только в JS. Именно поэтому JavaScript является самой популярной браузерной технологией [15].

Но есть еще одна особенность, которая делает JS еще более функциональнее: это интерфейсы прикладного программирования (API), которые предоставляют дополнительные возможности для использования в JavaScript.

API — это готовые наборы блоков кода, которые позволяют разработчику реализовывать программы, которые в противном случае было бы трудно или невозможно реализовать. Кроме встроенных API-интерфейсов в браузер, также существуют внешние API, которые разрабатываются группами разработчиков различных компаний. Благодаря им, можно реализовать интеграцию своей веб-страницы с их сервисами для удобства конечного пользователя [22].

Как было уже сказано, JavaScript обеспечивает полную интеграцию с HTML. А это означает, что чем больше возможностей появляется в HTML, тем функциональнее становится и JS. Как пример, HTML5 – эволюция стандарта HTML, добавляющая новые теги и, что более важно, ряд новых возможностей браузерам [12, 13].

Вот несколько примеров, что могут делать разработчики с помощью HTML5:

- Многозадачность с одновременным использованием нескольких ядер процессора.
- Проигрывание видео/аудио, без Flash.
- 2D и 3D-рисование с аппаратной поддержкой, как в современных играх.

Многие возможности HTML5 всё ещё в разработке, но браузеры частично обеспечивают их поддержку. Благодаря развитию HTML5 JavaScript становится все более мощным, что позволяет реализовывать дополнительный функционал [2].

Современные браузеры постоянно улучшают свои движки, чтобы увеличить скорость работы JS, правят баги и стараются следовать современным стандартам.

Очень важно то, что новые стандарты HTML5 и ECMAScript сохраняют максимальную совместимость с предыдущими версиями. Это позволяет JS полностью работать и на устаревших устройствах и версиях браузеров [23].

Но иногда разработчики браузеров стараются включить новые экспериментальные возможности, которые ещё не полностью описаны в стандарте. Со временем стандарт меняется, браузеры переписывают движок, и это может привести к ошибкам в уже написанном, основанном на старой реализации, JS-коде.

При этом все браузеры пытаются приблизиться к стандарту, и различий между ними становится все меньше и меньше, что обеспечивает универсальность JS-кода [18].

Вместе с JavaScript на страницах используются и другие технологии. Связка с ними может помочь добавить еще больше интерактивности при работе с пользователем.

Одна из таких технологий — Java. Это язык общего назначения, на котором можно писать самые разные программы. При работе с веб-страницам использую так называемые апплеты.

Апплет – это программа на языке Java, которую можно подключить к HTML при помощи тега `applet`:

```
<applet code="Biblioteka.class" codebase="/applets/">  
  <param name="views" value="50,30,70,20,40,60,80,35,65,75,85,90">  
</applet>
```

Такой тег загружает Java-программу из файла `Biblioteka.class` и выполняет её с параметрами `param`. Апплет выполняется в отдельной части страницы. Все действия пользователя внутри этого «контейнера» обрабатывает апплет. Контейнер может быть и скрыт, если нет необходимости показывать его пользователю [2].

Но чтобы апплет заработал, на компьютере посетителя веб-страницы должна быть установлена и включена среда выполнения Java, включая специальный плагин. Кроме всего этого, апплет еще должен быть подписан сертификатом, иначе Java заблокирует его и не будет выполнять из-за ограничений безопасности [36].

К достоинству апплетов можно отнести то, что Java имеет доступ к любым функциям компьютера пользователя, совсем как установленная программа [35].

Есть и недостатки:

- Java требует больше времени для загрузки;
- Среда выполнения Java, включая специальный плагин, должна быть установлена на компьютере посетителя и включена;
- Java-апплет не интегрирован с HTML-страницей, а выполняется отдельно. Но он может вызывать функции JavaScript;
- Java-апплеты должны быть подписаны сертификатом и получить согласие пользователя.

Увеличить интерактивность веб-страницы можно и с помощью специальных плагинов для браузера. Для этого можно использовать как JavaScript (Chrome, Firefox), так и язык C (ActiveX для Internet Explorer).

Эти плагины могут как отображать содержимое специального формата, так и взаимодействовать со веб-страницей.

Как и в ситуации с Java-апплетом, у них широкие возможности, но требуется их предварительная установка пользователем.

Не надо забывать и про Adobe Flash – кросс-браузерная платформа для мультимедиа-приложений, анимаций, аудио и видео. Flash-файл — это скомпилированная программа, написанная на языке ActionScript. Её можно подключить к HTML-странице и запустить в контейнере.

В первую очередь Flash полезен тем, что позволяет кросс-браузерно работать с микрофоном, камерой, с буфером обмена, а также поддерживает продвинутые возможности по работе с сетевыми соединениями.

Достоинства этой технологии:

- Сокеты, UDP для P2P и другие продвинутые возможности по работе с сетевыми соединениями
- Поддержка мультимедиа: изображения, аудио, видео. Работа с веб-камерой и микрофоном.

Однако есть и многочисленные недостатки:

- Flash должен быть установлен и включён. А на некоторых устройствах он вообще не поддерживается (например, все устройства от компании Apple).
- Flash не интегрирован с HTML-страницей, а выполняется отдельно.
- Существуют ограничения безопасности, однако они немного другие, чем в JavaScript.
- Имеет огромное количество уязвимостей.

Однако самый главный недостаток этой технологии заключается в прекращении ее поддержки производителем. Компания-разработчик Adobe

признала технологию устаревшей, поэтому жизненный цикл будет завершён в конце 2020 года [28].

В последние годы из-за того, что синтаксис JavaScript устраивает не всех, появилось много языков, которые добавляют различные возможности «поверх» JavaScript, а для запуска в браузере – при помощи специальных инструментов «трансляторов» превращаются в обычный JS-код. Это преобразование происходит автоматически и совершенно прозрачно, при этом неудобств в разработке и отладке практически нет [1].

При этом разные языки выглядят по-разному и добавляют совершенно разные вещи:

- Язык CoffeeScript сосредоточен на большей ясности и краткости кода.
- Язык TypeScript разрабатывает компанией Microsoft. Он сосредоточен на добавлении строгой типизации данных, которая позволяет упростить разработку и поддержку больших систем.
- Язык Dart разрабатывается компанией Google. Уникальным отличием Dart от других языков является то, что он имеет свою независимую среду выполнения, которая расширяет его возможности. Кроме того, среда выполнения доступна для встраивания в приложения, то есть работы вне браузера [27].
- Существуют также трансляторы, которые берут код, использующий возможности будущих стандартов JavaScript, и преобразуют его в более старый вариант, который понимают все браузеры. Например, babeljs.

Учитывая все эти достоинства и недостатки различных технологий, для создания интерактивной карты был выбран JavaScript.

1.2 Анализ веб-инструментов для создания интерактивных карт

На данный момент на рынке есть три главных проекта, которые за счет своей масштабности, популярности и детализации занимают лидирующие позиции в своей отрасли.

Это наш отечественный проект Яндекс.Карты, его зарубежный аналог Google Maps и открытый для любого OpenStreetMap.

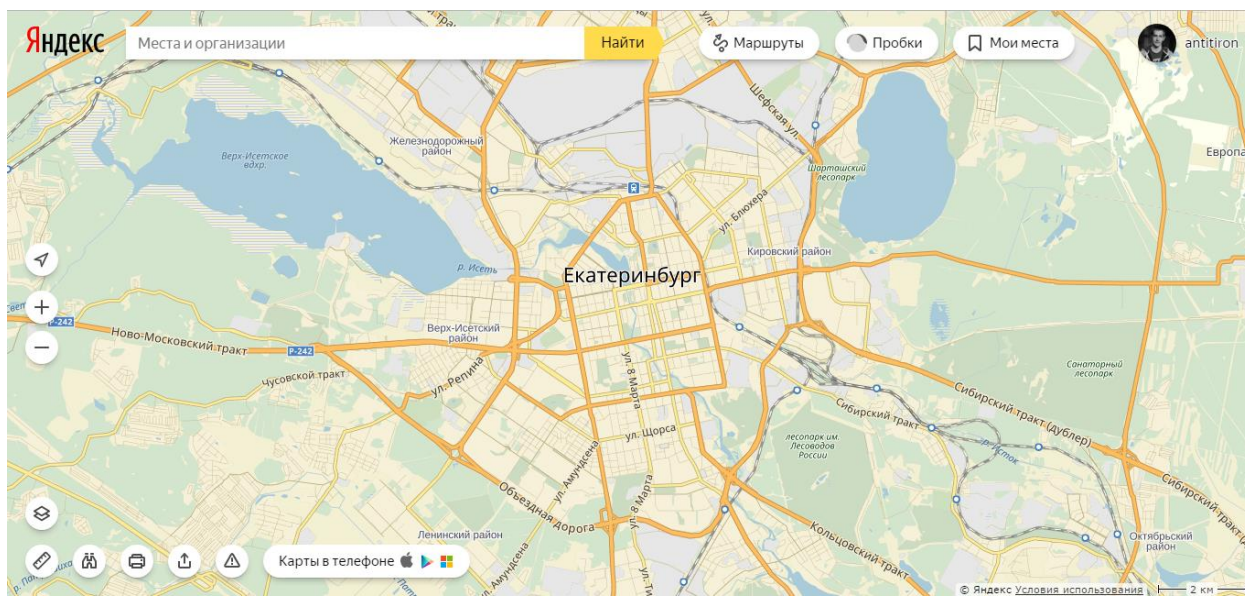


Рис 2 – Веб-интерфейс сервиса «Яндекс.Карты»

Яндекс.Карты — это поисково-информационная картографическая служба Яндекса, которая была открыта в 2004 году. Сервис представляет карты крупных городов и спутниковые снимки всего мира. С его помощью можно найти нужное место или организацию, узнать географические сведения об объекте, получить информацию о пробках на дорогах, спланировать свой маршрут, увидеть панорамы и фотографии улиц и многое другое.

JavaScript API Яндекс.Карт представляет собой набор JavaScript-компонентов, предназначенных для создания интерактивных карт. Все компоненты уже размещены на серверах Яндекса и доступны для использования сразу после их загрузки на страницу.

С помощью JavaScript API можно встроить на сайт или в приложение карту с поиском по топонимам и организациям, с возможностью строить

маршруты, смотреть панорамы и др. Картографические данные Яндекса постоянно обновляются, поэтому созданные с помощью API карты всегда актуальны — на них отображаются все изменения.

С помощью JavaScript API возможно настроить нужную логику взаимодействия пользователя с картой и определить, как эта карта будет выглядеть. Внешний вид объектов на карте, можно задать с помощью стандартных элементов или вовсе создать свой собственный макет.

API обеспечивает не только кросс-браузерность, но и мультиплатформность. Функционал созданной карты будет одинаковый на разных типах устройств.

JS API позволяет отобразить на карте большой объём информации или принять решение на основе анализа геоданных;

- Можно отобразить на карте большое количество объектов без потери производительности.
- Возможно использование вспомогательных методов, чтобы быстрее находить геообъекты на карте, определять их положение относительно друг друга и производить над ними другие операции.
- Можно отображать картографические данные на карте, загрузив файл в формате KML, GPX, YMapsML или JSON.
- Возможно создавать собственный слой карты, изменять проекцию или показать свои панорамы.

Так как JavaScript API Яндекс.Карт поддерживает обратную совместимость, написанный один раз код будет корректно работать с выходом новых версий.

Google Maps – это картографический сервис и технология, созданная компанией Google в 2005 году. Представляет собой карту и спутниковые снимки всего мира. Для многих регионов доступны высокодетализированные аэрофотоснимки. С сервисом интегрирован бизнес-справочник и карта автомобильных дорог с поиском маршрутов по всему миру.

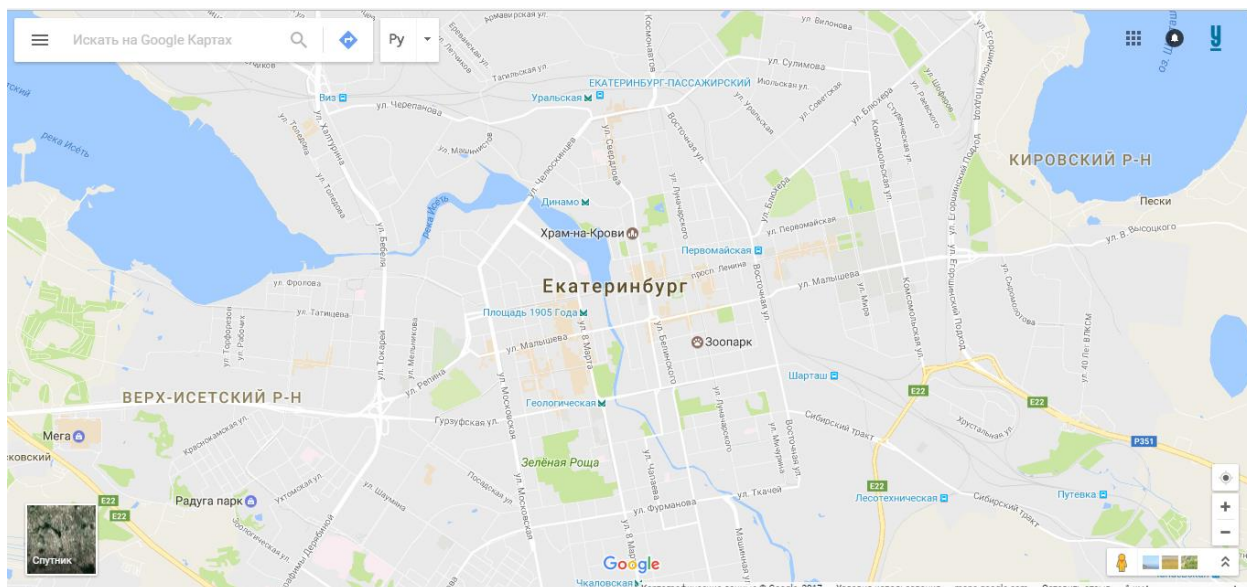


Рис 3 – Веб-интерфейс сервиса «Google Maps»

Существует возможность использовать сервис Google Maps в качестве основы для сторонних сервисов. Google Maps API — это коллекция инструментов для добавления карт Google в мобильные и веб-приложения, и создания на их основе многофункциональных решений для решения различных задач.

С помощью JavaScript API появляется возможность настраивать карты с собственным контентом и изображениями для отображения на веб-страницах и мобильных устройствах. JavaScript API Карт содержит четыре основных типа карт (дорожная карта, спутниковая, гибридная и рельефная), которые можно изменять с помощью слоев и стилей, элементов управления и событий, а также различных служб и библиотек [40].

Google постоянно обновляет и совершенствует свои данные, поэтому карты Google — одни из самых точных в мире при глобальном покрытии.

OpenStreetMap — некоммерческий веб-картографический проект по созданию силами сообщества пользователей Интернета подробной свободной и бесплатной географической карты мира.

Для создания карт используются данные с персональных GPS-трекеров, аэрофотографии, видеозаписи, спутниковые снимки и панорамы улиц,

предоставленные некоторыми компаниями, а также знания человека, рисующего карту. Лицензия OpenStreetMap позволяет свободно получать доступ ко всем растровым картам и всем лежащим в их основе пространственным данным, и этот проект направлен на поощрение нового и интересного использования этих данных.

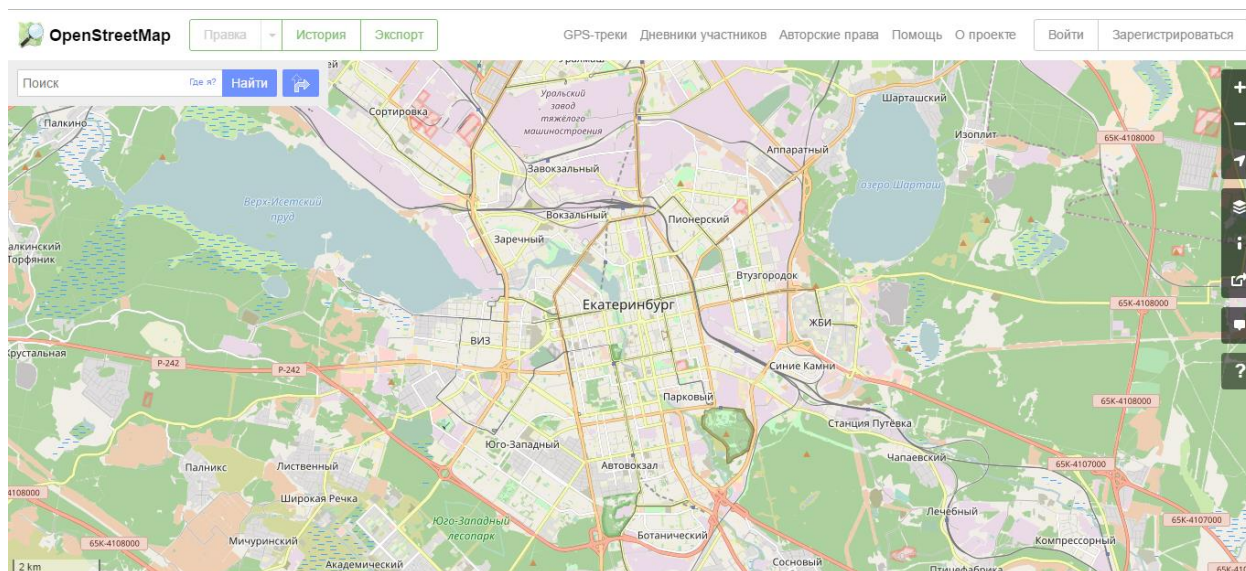


Рис 4 – Веб-интерфейс сервиса «OpenStreetMap»

В OpenStreetMap при создании карты используется принцип вики. Любой желающий может завести себе аккаунт и собственноручно редактировать карту. И если карты крупных городов в OpenStreetMap выглядят сопоставимо с картами Яндекс и Гугл, то в отображении дикой природы и сельской местности OpenStreetMap занимает лидирующую позицию [38].

Для OpenStreetMap также разработан API для редактирования, извлечения и сохранения необработанных геоданных из или в базу данных, благодаря чему возможно построение собственной карты со своими данными.

Для наглядности выбора составим сравнительную таблицу рассматриваемых веб-инструментов:

Таблица 1

Критерий/Сервис	Яндекс.Карты	Google Maps	OpenStreetMap
Покрытие	Лучшее в России, на втором месте по миру	Лучшее покрытие всего мира	Лучшие в мире, лучшие в России, может доходить вплоть до мелких деревушек, дачных поселков, мелких речушек, троп, но только при наличии мапперов, в остальных случаях конкуренты в разы лучше
Детализация	Хорошая детализация России, достаточная в мире	Хорошая детализация по всему миру. На карте России могут отсутствовать крупные города. В плане отображения невнятная детализация.	Одна из лучших детализаций в городах присутствия
Детализация на уровне здания	Крупные торговые центры	Крупные торговые центры	Есть, но встречается редко
Возможность загрузки и использования офлайн	Да. Большой размер данных, но только в рамках своих приложений	Да. Большой размер данных, но только в рамках своих приложений	Огромное количество всевозможных данных для разных типов устройств. Многочисленные утилиты для загрузки и работы с данными
Редактирование карт	Сообщение об ошибках, Народная карта	Сообщение об ошибках	Полная
Вариант выбора отображения ландшафта	Карта, спутник, гибрид	Карта, спутник, рельеф	Много разных вариантов, но нет снимков со спутников, где-то встречается вектор
Отображение пробок в крупных городах	Да, но только на территории России	Да, но не все города	Нет
Возможность общения между пользователями	Нет	Нет	Форум, личные сообщения, комментарии к изменениям

Критерий/Сервис	Яндекс.Карты	Google Maps	OpenStreetMap
Обзорные фотографии улиц	Яндекс Панорамы	Google Streetview	Нет
Поиск	Да. Интеллектуальный поиск	Да. Интеллектуальный поиск	Nominatum, но он достаточно нечеток
Голосовой ввод (на русском)	Да	Да	Нет
Режим 3D	Да	Да	Не ручная прорисовка
Ночной режим	Да	Да	Нет
Построение маршрута	Автомобиль, общественный транспорт, пешеходные маршруты. Строит с учетом пробок. Требуется интернет для построения	Автомобиль, общественный транспорт, пешеходный маршрут. Строит с учетом пробок и расписания общественного транспорта. Требуется интернет для построения	С помощью сторонних сервисов
Справочная информация	Подробная информация об организациях в России	Хуже других знает российские организации	POI, может быть не очень актуальная и далеко неполная
Актуализация гео-, справочной информации	Нет точной информации, но данные постоянно актуализируются	Нет точной информации, но данные постоянно актуализируются	Очень быстрое обновление гео-информации, но только в крупных городах, может не обновляться вообще
Возможность оставить отзывы и оценить организацию	Отзыв и оценка	Отзыв и оценка	Нет
Интерфейс и юзабилити	Современный интерфейс. Осуществление большинства функций возможно в два шага	Современный интерфейс. Некоторые функции не до конца понятны на интуитивном уровне	Имеет высокий порог вхождения. Некоторые функции имеют отвратительное юзабилити.
Итог	<i>Хорошо знает географию и организации по всей России. Наличие сервисов, «Панорамы улиц», голосового ввода, построения маршрута</i>	<i>Лучшие в мире по детализации. Удобное и функциональное построение маршрутов. Панорама улиц, голосовой ввод</i>	<i>Полное редактирование. Великолепная детализация в крупных городах. Огромное количество вариантов по работе с данными.</i>

Так как для нашего проекта необходима карта Уральского региона, то Google Maps не подошел из-за своей низкой детализации российских просторов.

От OpenStreetMap мы также отказались, так как, во-первых, это база данных, содержащей сведения о точках (а у портала «Наш Урал» уже есть собственная такая база) и любую оболочку пришлось бы писать самим с абсолютного нуля, а во-вторых, его превосходная детализация ограничивалась лишь теми городами, где присутствовали мапперы (люди, которые рисуют карты). А многие достопримечательности Урала расположены как раз там, где мапперов, к сожалению, нет. В этом случае выручают спутниковые карты, но у OpenStreetMap их просто нет.

В этом аспекте заметно выигрывает платформа Яндекс.Карты. В совокупности с хорошей детализацией карт уральского региона и прекрасным предоставляемым JavaScript API, для реализации интерактивной карты достопримечательностей Урала нами была выбрана именно эта платформа.

1.3 Техническое задание на разработку интерактивной карты

1.3.1 Общие требования

1. Необходимо провести анализ существующих интерактивных веб-технологий, выбрать наилучший вариант для создания интерактивной карты достопримечательностей Урала.

2. Создать базу данных достопримечательностей Урала, на основе статей портала «Наш Урал». В базе должны содержаться следующие сведения: наименование достопримечательности, фотография, gps-координаты, ссылка на статью. Все достопримечательности должны быть разбиты на группы. Данные описать в формате YMapsML.

3. Спроектировать и разработать интерактивную карту для построения туристических маршрутов по достопримечательностям Урала. Карта должна состоять из четырех разделов: шапка сайта, с логотипом и двумя кнопками (выбор готового маршрута и выбор региона Урала), меню с выбором группы достопримечательностей, основная часть с картой и подвал, где будет

отображаться информация о маршруте для пользователя (длина маршрута, время в пути, расход топлива и другая информация). На карточке достопримечательности необходимо вывести следующие данные: наименование достопримечательности, фотографию, gps-координаты. Добавить также три кнопки, для того чтобы закрыть карточку, открыть статью на сайте о достопримечательности, добавить достопримечательность в свой маршрут.

4. Протестировать карту на ошибки и опубликовать на портале «Наш Урал» по адресу (nashural.ru/map.htm);

5. Подготовить инструкцию по работе с картой для пользователей.

1.3.2 Требования к дизайну интерактивной карты Урала

Стилистическое оформление карты должно соответствовать текущему стилю портала «Наш Урал» и использовать его цветовые и графические элементы, а также логотип сайта. Использовать преимущественно синие оттенки. Шрифты, используемые в графических элементах карты, могут выбираться произвольно при условии не противоречия стилю портала.

Сайт должен корректно отображаться в всех современных браузерах, а также быть адаптивным.

1.3.3 Графические схемы шаблона карты



Рис 5 – Графическая схема интерактивной карты Урала

Наименование достопримечательности	
Фотография достопримечательности	
GPS-координаты достопримечательности	
Кнопка «Закрыть»	Кнопка «Ссылка на статью»
Кнопка «Добавить в маршрут»	

Рис 6 – Графическая схема карточки достопримечательности

1.3.4 Требования к функциональности интерактивной карты

Интерактивная карта должна позволять пользователям:

1. Перейти на главную страницу сайта «Наш Урал»;
2. Выбрать необходимые группы достопримечательностей и отобразить их на карте;
3. Отображать карточку достопримечательности с необходимыми данными;
4. Просматривать готовые маршруты из подготовленного списка;
5. Строить свои собственные маршруты, а также их редактировать;
6. Выбрать нужный регион Урала;
7. Просмотреть информацию о маршруте и скачать маршрутный лист себе на устройство.

ГЛАВА 2. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНТЕРАКТИВНОЙ КАРТЫ ДЛЯ ПОСТРОЕНИЯ ТУРИСТИЧЕСКИХ МАРШРУТОВ ПО ДОСТОПРИМЕЧАТЕЛЬНОСТЯМ УРАЛА

2.1 Проектирование интерактивной карты достопримечательностей Урала

Идея создания интерактивной карты достопримечательностей Урала возникла достаточно давно. В сети Интернет уже больше десяти лет существует портал «Наш Урал» - электронный путеводитель по достопримечательностям всего большого Урала. На нем представлены отчеты путешественников, справочные материалы, фото и видео галереи. Но больше всего он знаменит своей базой достопримечательностей. Каждая из них сопровождается таблицей, в которой указаны расстояния от крупных городов, а также присутствуют GPS-координаты для навигаторов.

При планировании самостоятельного путешествия возможность наглядно проследить, где находится достопримечательность, что еще интересного есть рядом, построить туристический маршрут и сохранить его, была бы очень полезна для пользователей портала.

В самом начале была разработана логика по взаимодействию пользователя с интерактивной картой. Посетитель посещают специальную страницу, на которой отображена карта достопримечательностей Урала, на которой с помощью специальных фильтров можно отобразить различные группы достопримечательностей Урала. После того, как пользователь настроил нужные ему фильтры, он может кликнуть по иконке достопримечательности, перед ним откроется карточка со всеми данными об этой достопримечательности. Эта информация, а также другие данные будут храниться в специальном файле в формате YMapsML.

После того, как пользователь открыл карточку достопримечательности, он может прочесть статью о ней, либо добавить в свой маршрут. Кроме того, с помощью специального разработанного функционала можно будет выбрать один из готовых маршрутов, для экономии времени посетителя.

Итак, наш проект будет состоять из следующих составляющих:

1. База данных достопримечательностей Урала в формате YMapsML;
2. Страница с веб-картой для взаимодействия с пользователем;
3. API Яндекс.Карт, который будет отображать описанные нами геоданные на карте и обрабатывать другие запросы, например, связанные с маршрутизацией.

Для создания дизайна интерфейса интерактивной карты будет использоваться уже готовые библиотеки, разработанные компанией Google: Material Design и dialog-polyfill. Для создания мобильной версии карты будут использованы услуги сервиса Phonegap. Это сделано для увеличения скорости разработки и сохранения удобства для пользователя [42].

Ниже представлена схема взаимодействия пользователя с интерактивной картой:



Рис 7 – Логика работы пользователя с картой

2.2 Описание процесса разработки интерактивной карты

Так как используемый веб-инструмент выполнен другой компанией, то перед началом работы с API необходимо внимательно ознакомиться с условиями лицензионного соглашения и условиями пользования, чтобы понять, а не нарушим ли мы их.

1. API должен использоваться только в проектах с открытым доступом (Бесплатная версия API предназначена только для сайтов или приложений, доступ к которым может получить любой пользователь интернета. Если есть регистрация, она должна быть доступной всем желающим) – проект полностью открыт. На данный момент регистрация в процессе разработки, но даже после ее внедрения, никаких ограничений на пользование сайтом без регистрации накладываться не будет.

2. Нет возможности использовать API в коммерческих проектах – проект некоммерческий.

3. Данные, полученные средствами API, должны быть показаны на карте Яндекса – все данные будут отображены на карте Яндекс.

4. Запрещается сохранять или изменять данные, полученные средствами API – изменять или сохранять данные не планируется.

5. Запрещается скрывать и видоизменять копирайты и логотип Яндекса на карте – нет необходимости.

6. Нет возможности использовать API для мониторинга и диспетчеризации – нет необходимости.

7. Нет возможности использовать API в игровых проектах – проект не входит в категорию «игровых».

8. Нет возможности превышать суточный лимит запросов к API (для бесплатной версии API лимит составляет 25 000 запросов в сутки) – на данный момент такой проблемы не стоит.

Итак, проект удовлетворяет всем требованиям для использования его в бесплатной версии.

Прежде чем заполнить интерактивную карту данными, необходимо создать базу данных достопримечательностей Урала. Для этого необходимо описать данные о достопримечательностях в формате YMapsML.

YMapsML (Yandex Maps Markup Language) — язык описания географических данных, разработанный в компании Яндекс. Под

географическими данными подразумеваются любые данные, географически привязанные к местности. [43]

YMapsML используется как формат передачи данных между сервисами Яндекс.Карт и сторонними программными средствами. Данные в формате YMapsML могут быть отображены на Яндекс.Картах, в этом же формате возвращает информацию о местоположении объектов сервис геокодирования.

YMapsML представляет собой открытый XML-формат. Для редактирования данных в формате YMapsML можно использовать любой текстовый редактор, а для их обработки — разнообразные программные средства, предназначенные для работы с XML.

Каждый элемент YMapsML содержит в себе один или несколько элементов. HTML-разметка всегда «экранируется», то есть отображается в браузере в виде текста. Исключение составляет HTML-код, содержащийся в элементах `gml:description` и `repr:text`, находящимся внутри `repr:Template`. Такой код интерпретируется браузером и выводится в форматированном виде.

Корневым элементом документа является элемент `ymaps:ymaps`. Он может содержать элементы `attr:Attribution`, один элемент `ymaps:GeoObjectCollection` и один элемент `repr:Representation`. Любой из этих элементов не обязателен.

Элемент `ymaps:GeoObjectCollection` представляет собой контейнер всех географических объектов и их коллекций. Для определения внешнего вида этих объектов, а также для задания параметров карты используется элемент `repr:Representation`. Элемент `attr:Attribution` содержит информацию о поставщиках сведений об организациях.

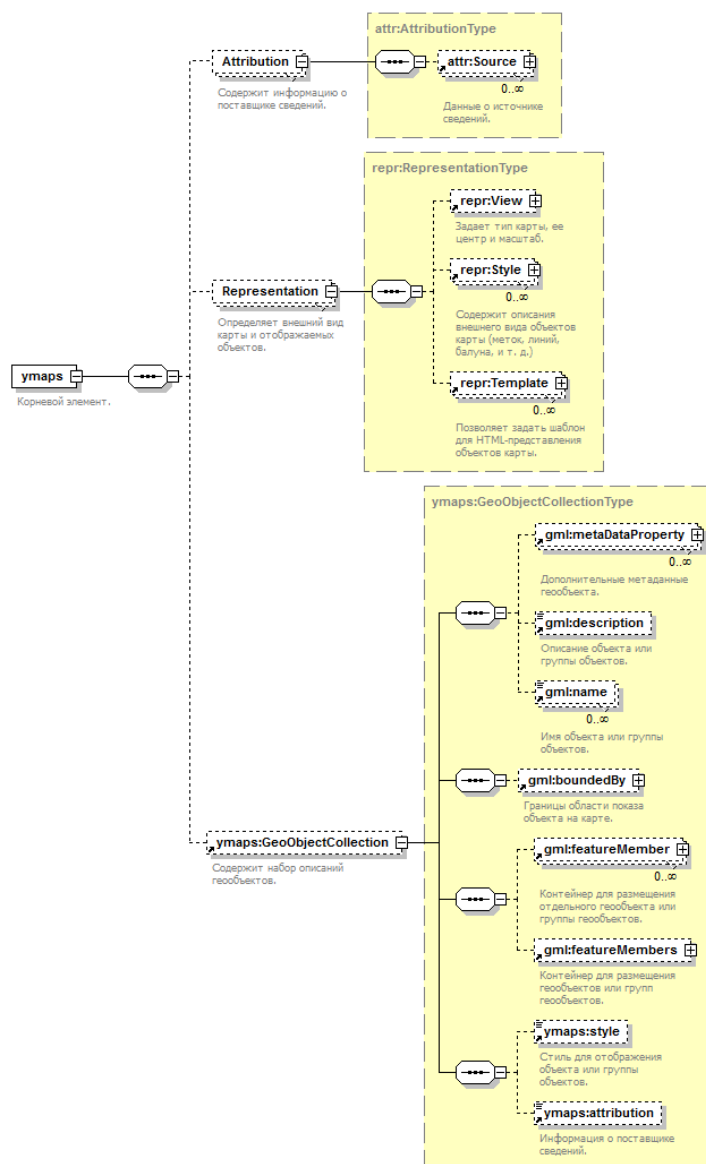


Рис 8 – Структура YMapsML-документа

Для описания географического объекта используется элемент `ymaps:GeoObject`. Чтобы обеспечить совместимость со стандартом GML каждый из этих элементов заключается в один из элементов-контейнеров — `gml:featureMember` или `gml:featureMembers`. Исключение составляет лишь элемент `ymaps:GeoObjectCollection`, находящийся в корневом элементе `ymaps:ymaps`.

Разница между этими контейнерами заключается в следующем. Элемент `gml:featureMember` может содержать либо описание одного географического объекта (`ymaps:GeoObject`), либо одной коллекции

(ymaps:GeoObjectCollection). Элемент gml:featureMembers может содержать в себе произвольное количество описаний географических объектов и их коллекций.

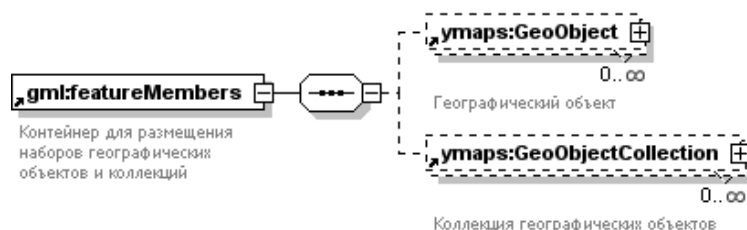


Рис 9 – Структура элемента gml:featureMembers

Географическому объекту можно сопоставить следующие данные:

1. Метаданные.
2. Геометрические и географические данные.
3. Информация о визуальном отображении географического объекта.
4. Информация о поставщике сведений об организации.

Структура элемента ymaps:GeoObject, содержащего описание географического объекта, выглядит следующим образом.

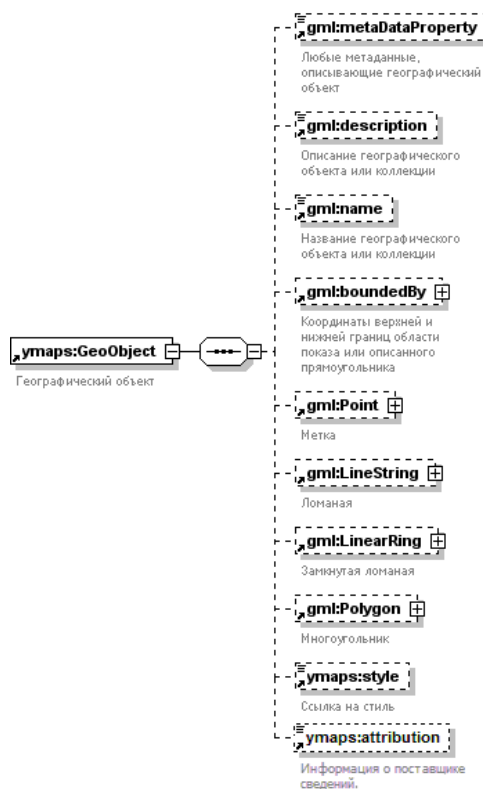


Рис 10 – Структура элемента ymaps:GeoObject

С помощью коллекций можно объединять как географические объекты, так и сами коллекции. Ниже представлена структура элемента `ymaps:GeoObjectCollection`.

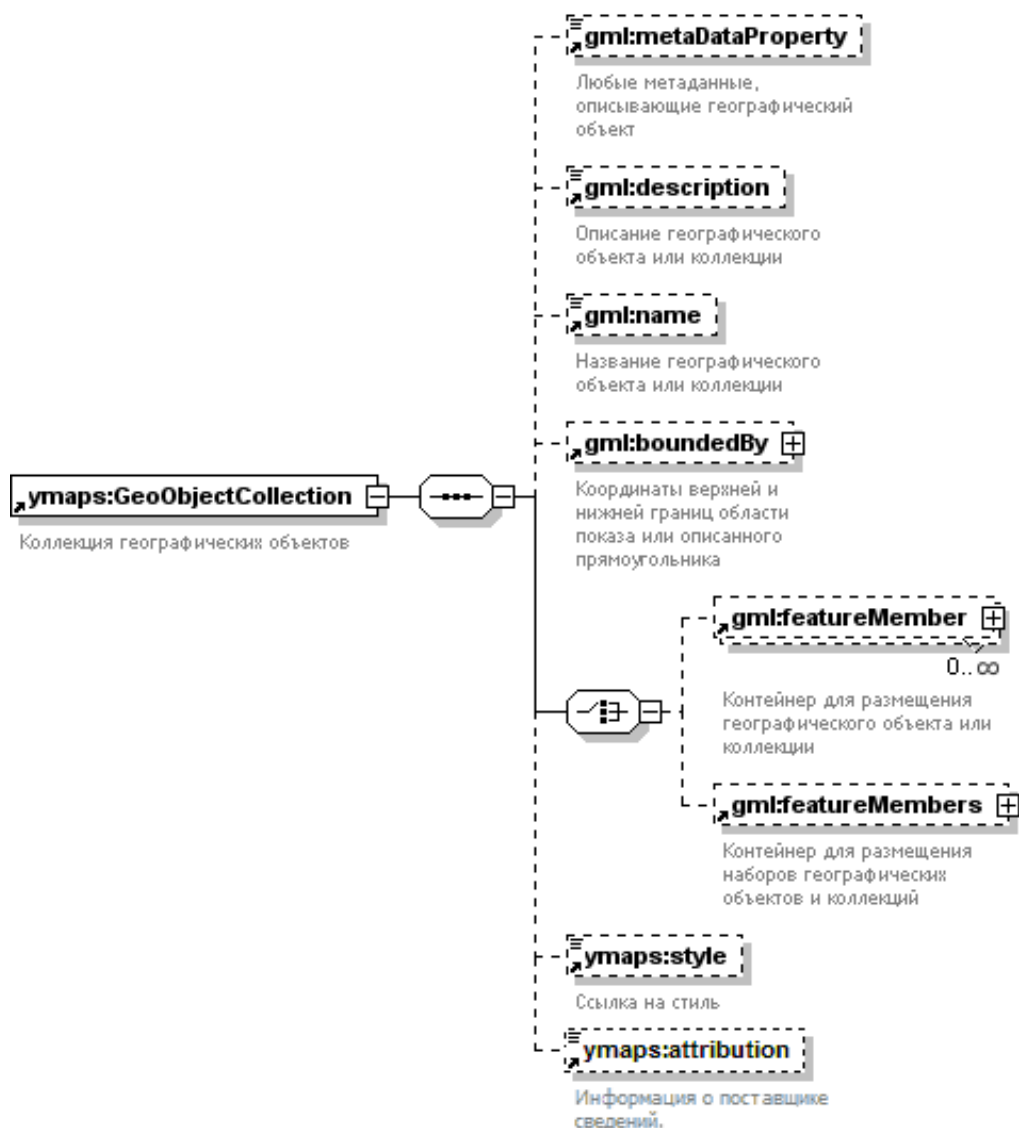


Рис 11 – Структура элемента `ymaps:GeoObjectCollection`

При работе с географическими объектами и их коллекциями необходимо дать им названия, описания, указать другую дополнительную информацию. Такие данные называют метаданными. Другими словами, это любые данные, описывающие географические объекты и их коллекции и не относящиеся к геометрическим и географическим данным [8].

Для размещения метаданных в YMapsML предусмотрены следующие элементы:

- `gml:name` — название географического объекта или коллекции;
- `gml:description` — описание географического объекта или коллекции. Может содержать HTML-код.
- `gml:metaDataProperty` — любые другие метаданные в XML-формате, описывающие географический объект или коллекцию. Данные размещаются во вложенном элементе `ymaps:AnyMetaData`.

Для описания геометрических свойств объектов и их привязки к географическим координатам используются следующие элементы:

- `gml:Point` — геометрическая точка. Используется в том случае, когда форма и размеры географического объекта неважны. Местоположение точки задаётся географическими координатами.
- `gml:LineString` — ломаная линия. Определяется с помощью задания географических координат её вершин.
- `gml:LinearRing` — замкнутая ломаная линия. Определяется с помощью задания географических координат её вершин. Автоматически достраивается до замкнутой линии путём соединения последней и первой вершин.
- `gml:Polygon` — многоугольник. Определяется с помощью задания географических координат вершин ломаных линий, являющихся границами многоугольника. Все ломаные автоматически достраиваются до замкнутых путём соединения первых и последних вершин.

В интерактивной карте будут использовать только географические точки. На данный момент нет необходимости и данных для описания форм и размеров достопримечательностей Урала. В этом случае координаты геометрической точки указываются в элементе `gml:pos`.

```
<ymaps:GeoObject>
```

```
  <gml:name>Дворец культуры Нижней Салды</gml:name>
```

```

    <gml:metaDataProperty>
      <ymaps:AnyMetaData>
        <href>//nashural.ru/mesta/sverdlovskaya-oblast/top-17-
dostoprimechatelnostej-nizhnej-saldy/</href>
        <src>//nashural.ru/assets/uploads/DSC_5785-
e1551210290833.jpg</src>
      </ymaps:AnyMetaData>
    </gml:metaDataProperty>
    <gml:Point>
      <gml:pos>60.715073 58.071218</gml:pos>
    </gml:Point>
  </ymaps:GeoObject>

```

YMapsML позволяет задавать внешний вид отображаемых на карте объектов с помощью стилей. Стили позволяют задавать параметры отображения географических объектов и создавать уникальное HTML-представление с помощью шаблонов.

Для определения стилей используется элемент `repr:Style`, шаблонов — элемент `repr:Template`. Чтобы указать, какой стиль следует применять при отображении географического объекта или коллекции, следует использовать элемент `ymaps:style`.

```

<ymaps:GeoObject>
  <ymaps:style>#Style</ymaps:style>
  <gml:Point>
    <gml:pos>60.715073 58.071218</gml:pos>
  </gml:Point>
</ymaps:GeoObject>

```

Здесь `#Style` — ссылка на идентификатор элемента `repr:Style`, содержащий описание стиля.

При описании геообъекта или коллекции необходимо ссылаться только на стиль. Ссылка на шаблон (элемент `repr:Template`) не сработает.

Каждому типу географических объектов соответствует собственный элемент YMapsML (потомок элемента `repr:Style`), с помощью которого задаются параметры его отображения. В нашей карте будут использоваться метки геообъектов, поэтому задавать параметры метки необходимо в элементе `repr:iconStyle` или `repr:iconContentStyle`.

YMapsML поддерживает наследование стилей. Это позволяет использовать в стиле параметры и шаблоны, ранее определённые в другом стиле. Для указания наследуемого стиля используется элемент `repr:parentStyle`.

```
<repr:Style gml:id="mountains">
  <repr:iconStyle>
    <repr:href>/map/icons/mountains.png</repr:href>
  </repr:iconStyle>
</repr:Style>
```

Шаблоны позволяют сформировать произвольное HTML-представление геообъектов, балунов и хинтов. Например, если для отображения метки требуется использовать изображение, текст, а также другие метаданные, то необходимо сконструировать соответствующий шаблон и задействовать его в стиле, использующемся для отображения метки.

Для определения шаблона используется элемент `repr:Template`, внутри которого в контейнере `repr:text` размещается HTML-код (тело шаблона).

```
<repr:balloonContentStyle>
  <repr:template>#balloonTemplate</repr:template>
</repr:balloonContentStyle>
<repr:Template gml:id="balloonTemplate">
  <repr:text>
    <![CDATA[<div style="font-weight:bold;padding-
      bottom:5px;color:#ff0000">${name}</div>
    <div align="center">
      <a href="https:${metaDataProperty.AnyMetaData.href}"
        target="_blank">
```

```
<img src="https:${metaDataProperty.AnyMetaData.src}</img>
</a></div>]]>
```

```
</repr:text>
```

```
</repr:Template>
```

После создания YMapsML-файла достопримечательностей Урала, необходимо его подключить к интерактивной карте. Для загрузки используется метод `load`. После того, как данные загрузятся, метод `load` преобразует их в объект `GeoObjectCollection` и передает ссылку на этот объект функции-обработчику.

```
ymaps.geoXml.load('https://nashural.ru/data.xml?v1.0.9.1')
  .then(function(res) {
    map = new ymaps.Map('map', {
      center: [56.80671065838881, 60.54288274121081],
      zoom: 8,
      controls: [buttonEditor, 'geolocationControl', 'typeSelector',
'zoomControl']], {buttonMaxWidth: 300}
    );
    res.geoObjects.each(addMenuItem);
    $('#map-spinner').remove();
  })
  .catch(function(error) {
    console.error(error);
    throw error;});
```

Базовой системой координат JavaScript API является глобальная пиксельная система координат. Любая координатная система реализуется с помощью проекции. Проекция устанавливает соответствие между географическими координатами точки и глобальными пиксельными координатами для каждого уровня масштабирования. [41]

API включает в себя встроенную поддержку стандартных географических координат и декартовой системы. Для географических

координат API предоставляет две проекции: эллиптическую (согласно WGS-84) и сферическую проекции Меркатора. Первая используется в API Яндекс.Карт по умолчанию, вторая часто применяется в сторонних географических сервисах [7].

При создании карты ей ставится в соответствие проекция, которая в дальнейшем не может быть изменена. Координаты всех объектов, размещаемых на карте, пересчитываются в глобальные пиксельные координаты в соответствии с этой проекцией.

Тип карты можно указать в конструкторе или с помощью метода setType. Можно использовать как встроенные, так и собственные типы карт.

Необходимо добавить на карту переключатель типа карты, кнопки изменения масштаба и определения геолокации пользователя. Кроме того, нужно использовать свою дополнительную кнопку для редактирования существующего маршрута, поэтому ее нужно создать заранее:

```
controls: [buttonEditor, 'geolocationControl', 'typeSelector', 'zoomControl']],  
{buttonMaxWidth: 300}
```

Теперь необходимо составить легенду карты, заполнив ее данными из сформированного YMapsML-файла.

```
var icon_link =  
group.properties.get('metaDataProperty.AnyMetaData.icon_link');  
var name = group.properties.get('name');  
$('<a>')  
  .text(name)  
  .attr('href', 'javascript: void(null)')  
  .addClass('mdl-navigation__link')  
  .addClass(icon_link)  
  .on('click', onclick)  
  .appendTo($mapGroups);  
function onclick(event) {  
  var $this = $(this);
```



```

var isActive = $this.hasClass('is-active');
if (isActive) {
    map.geoObjects.remove(group);
} else {
    map.geoObjects.add(group); }
$this.toggleClass('is-active', !isActive);
if (isMobile()) {
document.querySelector('.mdl-layout').MaterialLayout.toggleDrawer() }
event.preventDefault();}}

```

Функция получения информации о достопримечательности и вывод ее карточки пользователю:

```

group.events.add('click', function(event) {
    //выбираем достопримечательность из базы по клику
    var target = event.get('target');
    //сохраняем ее имя
    var name = target.properties.get('name');
    dname = name;
    //ссылку на сайте
var href = target.properties.get('metaDataProperty.AnyMetaData.href');
    dhref = href;
    //картинку
    var src = target.properties.get('metaDataProperty.AnyMetaData.src');
    pos = target.geometry.getCoordinates();
    //получаем и выводим gps-координаты прямо в карточке
достопримечательности
    $(".gps").empty();
    $(' .gps').append("<b>GPS: </b> " + pos);
    //создаем карточку достопримечательности и выводим всю
полученную информацию
    var $spinner = $('<div></div>')

```

```

.addClass('mdl-spinner mdl-js-spinner is-active')
.css({
    position: 'absolute',
    top: 'calc(50% - 14px)',
    left: 'calc(50% - 14px)'
});
var $img = $('<img>')
.attr('src', src)
.on('load', function() {
    $('.mdl-dialog__content', dialog)
        .empty()
        .append($img);
});
$('.mdl-dialog__title', dialog).text(name);
$('.mdl-dialog__content', dialog)
    .empty()
    .append($spinner);
$('.js-dialog-ok', dialog).data('href', href);
if (!dialog.open) {
    dialog.showModal();} });

```

Прежде чем использовать функции API Яндекс.Карт на веб-странице, необходимо загрузить в браузер JavaScript-файлы, в которых эти функции определены [21, 20]. Для этого нужно добавить в футер (подвал) HTML-страницы строку следующего вида:

```

<script src="https://api-maps.yandex.ru/2.1/?lang=ru-
RU&apikey=8021680117ab"></script>

```

По адресу `api-maps.yandex.ru/2.1/` находится загрузчик, который при обращении к нему подключает нужные компоненты API. С недавнего времени Яндекс также требует от разработчиков получения API-ключа, чтобы получать точную статистику по потреблению ресурсов [37].

Так как нашей интерактивной картой будут в основном пользоваться люди из России, то необходимо установить по умолчанию русскую локализацию карты с помощью параметра `lang` и значения `ru-RU`.

Всю страницу надо разграничить на четыре части: каталог достопримечательностей слева, главное меню сверху, информационное поле снизу и сама карта в оставшейся части страницы.

Создадим видимый контейнер ненулевого размера, в котором будет размещена карта. В качестве контейнера можно использовать любой HTML-элемент блочного типа (например, элемент `div` с заданной шириной и высотой). Карта заполнит этот элемент полностью [16].

```
<body><div id="map" style="position: absolute; width: 100%; height: 100%"></div></body>
```

Для создания карты предназначен класс `Map`. В конструкторе класса необходимо указать центр и коэффициент масштабирования карты и HTML-элемент, в котором она будет размещена.

Для указания HTML-элемента можно использовать как ссылку на элемент, так и его идентификатор (значение атрибута `id`) [29].

Необходимо также проработать возможность выбора определенного региона Урала. Надо создать выпадающее меню со списком регионов Урала. При нажатии на выбранный регион карта будет автоматически центрироваться по координатам соответствующей столицы.

Создадим список с указанием класса, чтобы получать данные для скрипта, которые будет автоматически формировать список регионов.

```
<ul class="mdl-menu mdl-menu--bottom-left mdl-js-menu mdl-js-ripple-effect" for="demo-menu-lower-left">
```

Необходимо указать название области, координаты для центрирования карты (столица области), а также значение параметра `zoom`, чтобы отдалять карту на расстояние по умолчанию [39].

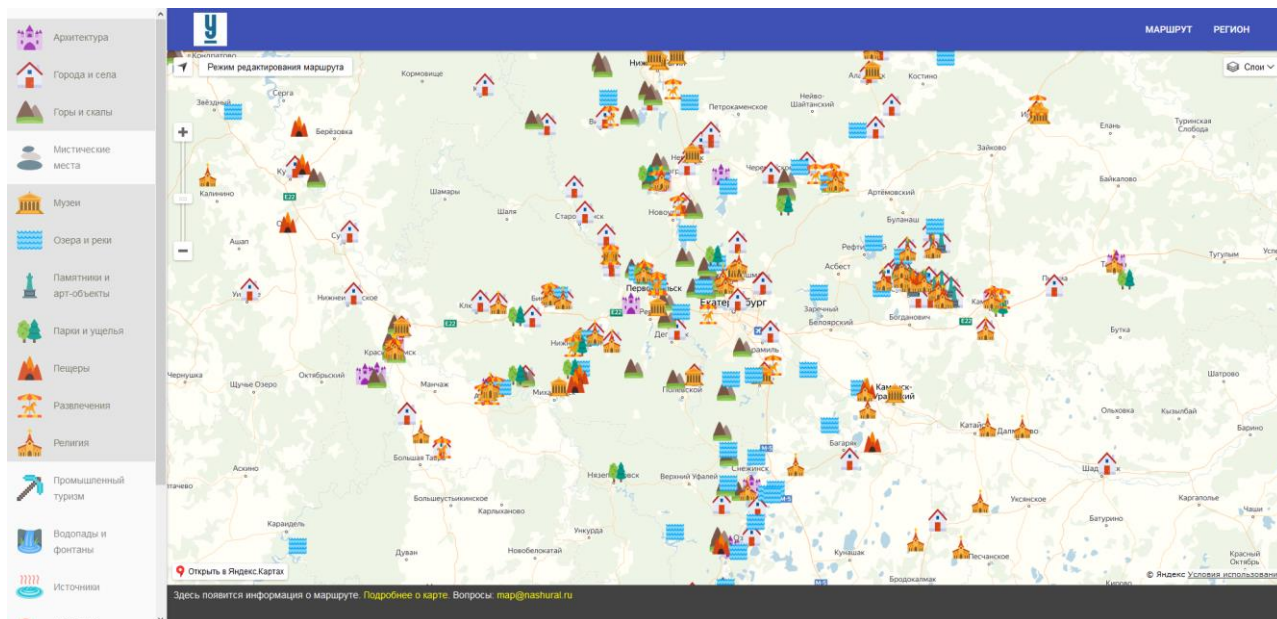


Рис 12 – Интерактивная карта достопримечательностей Урала

```
<li class="mdl-menu__item js-map-center" data-map='{ "center":  
[56.788798012365895,60.60339449999997], "zoom": 8}'>Свердловская  
область</li>
```

```
<li class="mdl-menu__item js-map-center" data-map='{ "center":  
[55.15336244294198,61.39169750000001], "zoom": 8}'>Челябинская  
область</li>
```

```
<li class="mdl-menu__item js-map-center" data-map='{ "center":  
[66.57161669007381,66.61538300000002], "zoom": 8}'>Ямало-Ненецкий  
АО</li></ul>
```

С помощью метода `setCenter()` нужно установить центр и коэффициент масштабирования карты. Сам центр задается в географических координатах.

```
$(document).on('click', '.js-map-center', jsMapCenter);
```

```
function jsMapCenter(event) {  
    var data = $(event.target).data('map');  
    map.setCenter(data.center, data.zoom);}
```

Внешний вид карты будет настроен с помощью CSS, поэтому необходимо подключить файл со стилями к веб-странице [4].

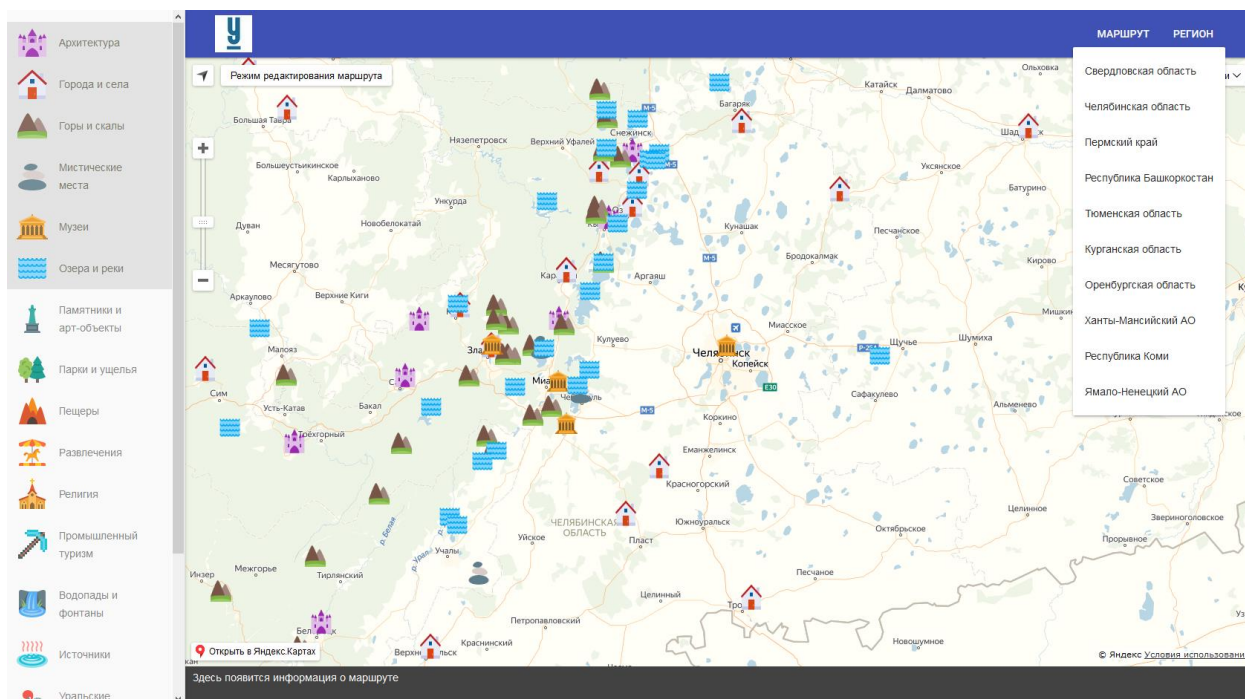


Рис 13 – Список регионов Урала

После создания интерактивной карты надо поработать над созданием функционала создания маршрутов пользователями. Необходимо, чтобы пользоваться картой могли, как опытные туристы, так и начинающие путешественники. Поэтому будет проработано два варианта создания своего маршрута: самостоятельное добавление в путевой лист достопримечательностей из базы данных и готовые маршруты из списка самых популярных по версии портала «Наш Урал».

Яндекс.Карты предоставляет свой собственный сервис Маршрутизатор. Сервис позволяет автоматически вычислять маршрут между заданными пунктами и получать информацию о проложенном маршруте (длина и время в пути). Маршруты можно строить как с учетом пробок, так и без [24].

Маршрутизатор Яндекс.Карт позволяет строить пять типов маршрутов: автомобильные, пешеходные, на общественном транспорте, на велосипеде и мультимаршруты (они включают в себя все другие типы маршрутов).

Для заданных точек API строит несколько оптимальных маршрутов. Самый быстрый маршрут устанавливается в качестве активного. Он отображается на карте фиолетовым цветом. Альтернативные маршруты

отображаются серым цветом. По клику пользователи смогут переключаться между маршрутами [9].

Построить маршрут на карте можно двумя способами.

Способ первый: через Панель маршрутизации. Панель маршрутизации позволяет пользователям задать начальную и конечную точки маршрута, выбрать необходимый тип маршрутизации или удалить построенный маршрут с карты. Программно можно лишь задавать точки маршрута и выбирать тип маршрутизации.

Преимущества этого метода:

- Удобный интерфейс: пользователи сами управляют маршрутом.
- Позволяет строить маршруты с возможностью вызова такси.

Недостатки:

- Не поддерживает промежуточные точки — маршруты строятся только по двум точкам.

Способ второй: на уровне кода, с помощью класса `multiRouter.MultiRoute`. Класс `multiRouter.MultiRoute` позволяет строить составные маршруты — маршруты по трем точкам и более. С помощью этого класса можно получать всю доступную информацию о маршруте, изменять внешний вид построенного маршрута и переставлять точки маршрута [32].

Преимущества:

- Позволяет строить составные маршруты по неограниченному числу точек.
- Позволяет строить маршруты на велосипеде.

Недостатки:

- Более сложная реализация.

С учетом этих преимуществ и недостатков, был выбран вариант построения маршрутов с помощью класса `multiRouter.MultiRoute`. Потому что только этот способ позволяет строить маршруты для трех точек и более, делая маршруты масштабнее, интереснее без каких-либо ограничений.

Перед началом работы разберемся с основными понятиями при работе с маршрутизатором.

Мультимаршрут — базовый объект для работы с маршрутизацией. Он содержит информацию о построенных маршрутах: геометрию, данные о точках, время в пути и т.д. Мультимаршрут обрабатывает эти данные и создает представление для отображения маршрутов на карте.

Активный маршрут — основной маршрут. Отображается на карте фиолетовым цветом. По умолчанию — самый быстрый маршрут из числа построенных [3, 20].

Путевая точка (или точка остановки) — точка маршрута, в которой нужно сделать остановку. К путевым точкам также относятся начальная и конечная точки.

Транзитная точка — промежуточная точка, через которую нужно проложить маршрут, но в которой не подразумевается остановка.

Путь — линия маршрута, которая соединяет путевые точки.

Сегмент — это отрезки маршрута между:

- путевыми или транзитными точками;
- точками возможного изменения направления движения (развилка, въезд, съезд, поворот, разворот, перекресток);
- точками, в которых меняется тип участка. Например, сегмент «пешком» меняется на сегмент «на автобусе».

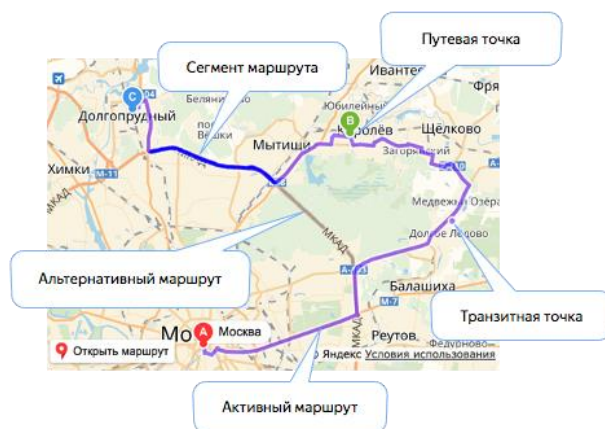


Рис 14 – Основные элементы маршрута на карте

Чтобы построить маршрут с помощью класса `multiRouter.MultiRoute`, необходимо создать экземпляр этого класса и добавить его на карту.

Перед созданием класса `multiRouter.MultiRoute` необходимо объявить переменные, которые будут необходимы для дальнейшей работы:

```
//храним построенный маршрут
var route;

//храним gps-координаты достопримечательностей
var pos;

//храним название достопримечательностей
var dname = "";

//храним ссылку на статью
var dhref = "";

//храним маршрутный лист пользователя
var marshrut = 'Маршрутный лист (сгенерирован порталом «Наш Урал»
по ссылке https://nashural.ru/map.htm):' + '\r\n\r\n';

//храним координаты маршрута по выбранным
достопримечательностям
var userroute = [];
```

Сразу стоит уточнить, что по умолчанию создается маршрут, который невозможно редактировать. При добавлении маршрута на карту режим редактирования выключен — точки маршрута являются неактивными, и пользователи не могут их двигать. Включить режим редактирования можно с помощью метода `start()`:

```
//объявляем и создаем кнопку на карте для редактирования маршрутов
var buttonEditor = new ymaps.control.Button({
    data: { content: "Режим редактирования маршрута" }});

//включение режима редактирования
buttonEditor.events.add("select", function () {
    route.editor.start({
        addWayPoints: true,
```



```

        removeWayPoints: true});});
// Выключение режима редактирования.
buttonEditor.events.add("deselect", function () {
    route.editor.stop();});

```

Рассмотрим процесс создания готового маршрута. Необходимо заготовить координаты путевых точек маршрута, хранить его описание и ссылку на файл с маршрутом, чтобы пользователь мог его скачать и добавить в навигатор.

Нужно добавить кнопку на карту со списком маршрутов, а также указать все его свойства:

```

//кнопка для списка готовых маршрутов
<button id="demo-menu-lower-right-left" class="mdl-button mdl-js-button
mdl-button--header">Маршрут</button>
//список маршрутов
<ul class="mdl-menu mdl-menu--bottom-right mdl-js-menu mdl-js-ripple-
effect" for="demo-menu-lower-right-left">
    <li class="mdl-menu__item js-map-route" data-map='[[58.010450,
56.229434], [60.155265, 56.250662], [60.081385, 56.335759], [60.457576,
56.459565], [60.466692, 56.449469], [60.490661, 56.368296]]' data-
url='https://nashural.ru/article/travel/samaya-trudnodostupnaya-
dostoprimechatelnost-rossii-i-ozero-mertvoj-vody/' data-
file='https://nashural.ru/map/marshrut/ozero-mertvoy-vody.txt'>Озеро мертвой
воды</li>
    <li class="mdl-menu__item js-map-route" data-map='[[58.010450,
56.229434], [60.152916, 59.743733], [60.194678, 58.848953], [60.166944,
58.8025], [60.2011, 58.725]]' data-
url='https://nashural.ru/article/travel/zhigalanskie-vodopady-i-plato-kvarkush/'
data-file='https://nashural.ru/map/marshrut/zhigalan.txt'>Жигаланские
водопады</li></ul>

```

Массив с точками маршрута хранится в атрибуте data-map, в атрибуте data-url хранится ссылка на маршрут на сайте, а в атрибуте data-file – готовый маршрутный лист со списком точек и их названиям. [34]

После этой подготовки необходимо функцию построения готового маршрута на карте:

```
function jsMapRoute(event) {  
    //удаляем другие маршруты с карты  
    map.geoObjects.remove(route);  
    //получаем координаты маршрута  
    var data = $(event.target).data('map');  
    //получаем ссылку с сайта на маршрут  
    var url = $(event.target).data('url');  
    //получаем файл на маршрут  
    var file = $(event.target).data('file');  
    //строим маршрут по координатам  
    var multiroute = new ymaps.multiRouter.MultiRoute({referencePoints:  
data}, {boundsAutoApply: true});  
    //сохраняем маршрут, чтобы можно было его удалить  
    route = multiroute;  
    //добавляем маршрут на карту  
    map.geoObjects.add(multiroute);}
```

Объект multiRouter.MultiRoute содержит информацию о построенных маршрутах. Например, протяженность и время преодоления, а также информацию о перекрытых участках на автомобильном маршруте. Чтобы получить ссылку на активный маршрут, нужно воспользоваться методом getActiveRoute().

//проверяем, что маршрут успешно создан, получаем и выводим информацию о нем пользователю

```
multiroute.model.events.add('requestsuccess', function() {  
    var activeRoute = multiroute.getActiveRoute();
```

```

var benzin = Math.round (activeRoute.properties.get("distance").value
/ 1000 / 10);

$(".dlina").empty();

$('.dlina').append("<b>Длина маршрута:</b> " +
activeRoute.properties.get("distance").text);

$(".benzin").empty();

$('.benzin').append("<b>Расход топлива:</b> примерно " + benzin +
" литров");

$(".urltravel").empty();

$('.urltravel').append("<b>Маршрут:</b> <a href='" + url +
">прочитать</a> или <a download href='" + file + ">скачать</a>");

$(".timetravel").empty();

$('.timetravel').append("<b>Время в пути:</b> " +
activeRoute.properties.get("duration").text);});

```

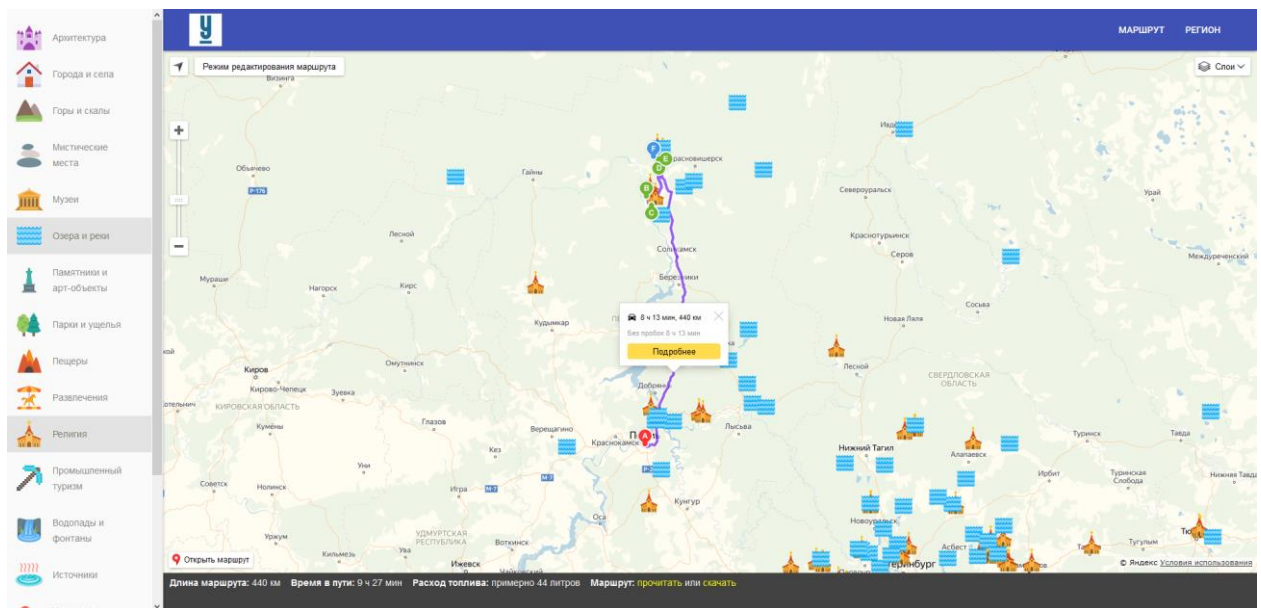


Рис 15 – Готовый маршрут на карте

Теперь нужно приступить к более сложной задаче: построение самостоятельных путевых листов. Необходимо, чтобы пользователь мог выбрать любую из достопримечательностей, добавить ее в маршрут, смог

увидеть длину маршрута, примерное время в пути, расход топлива и, главное, скачать путевой лист со всеми координатами. Чтобы помочь пользователю, в этот путевой лист нужно добавить еще и ссылки на достопримечательности, которые он выбрал – чтобы можно было подробнее их изучить.

Сама функция будет похожа на предыдущую, но уникальность этой функции будет заключаться в том, что она будет формировать файл с содержимым для пользователя с помощью возможностей JavaScript. Для этого будет использоваться метод `btoa()`, предназначенный для кодирования строки. Принцип работы метода заключается в том, что он сначала преобразует строку в последовательность бит, которые затем на основе схемы base-64 преобразуются в закодированную строку [30].

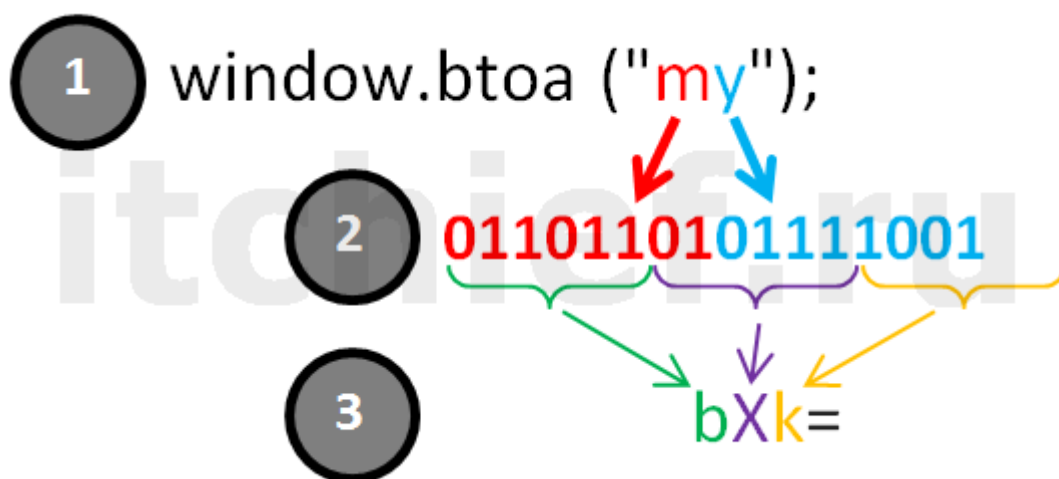


Рис 16 – Пример работы метода `btoa()`

Стоит сразу оговориться, что если в качестве параметра метода `window.btoa()` будет использоваться строка Unicode, то может произойти исключение, из-за того что данная строка может содержать символы вне диапазона и мы получим ошибку [13].

Чтобы этого не допустить, необходимо использовать следующий паттерн: `window.btoa(unescape(encodeURIComponent(str)))`. В этом случае, метод `unescape(str)` создает новую строку, в которой шестнадцатичная последовательность символов заменяется эквивалентами из кодировки

ASCII, а метод encodeURIComponent кодирует URI (англ. Uniform Resource Identifier – унифицированный идентификатор ресурса) заменяя каждый экземпляр определенных символов одной, двумя, тремя или четырьмя управляющих последовательностей, представляющими кодировку символа UTF-8 [25].

Таким образом, функция по созданию самостоятельных маршрутов пользователем будет иметь следующий вид:

```
function AddRoute(event) {  
    //добавляем координаты достопримечательности в путевой лист  
    useroute.push(pos);  
    marshrut = marshrut + pos + ' — ' + dname + ' (https:' + dhref + ')' + '\r\n';  
    //удаляем другие маршруты с карты  
    map.geoObjects.remove(route);  
    //строим маршрут по координатам  
    var multiroute = new ymaps.multiRouter.MultiRoute({referencePoints:  
useroute});  
    //сохраняем маршрут, чтобы можно было его удалить  
    route = multiroute;  
    //добавляем маршрут на карту  
    map.geoObjects.add(multiroute);  
    //получаем и выводим информацию о маршруте в консоль  
    multiroute.model.events.add('requestsuccess', function() {  
        var activeRoute = multiroute.getActiveRoute();  
        var benzin = Math.round (activeRoute.properties.get("distance").value / 1000  
/ 10);  
        $(".dlina").empty();  
        $('dlina').append("<b>Длина маршрута:</b> " +  
activeRoute.properties.get("distance").text);  
        $(".benzin").empty();
```

```

$('.benzin').append("<b>Расход топлива:</b> примерно " + benzin + "
литров");

$('.timetravel').empty();

$('.timetravel').append("<b>Время в пути:</b> " +
activeRoute.properties.get("duration").text);

var type = 'data:application/octet-stream;base64, ';
var base = btoa(unescape(encodeURIComponent(marshrut)));
var res = type + base;

$('.urltravel').empty();

$('.urltravel').append("<b><a download='nashural.ru.txt' href='" + res +
'">Скачать маршрут</a></b>");}}

```

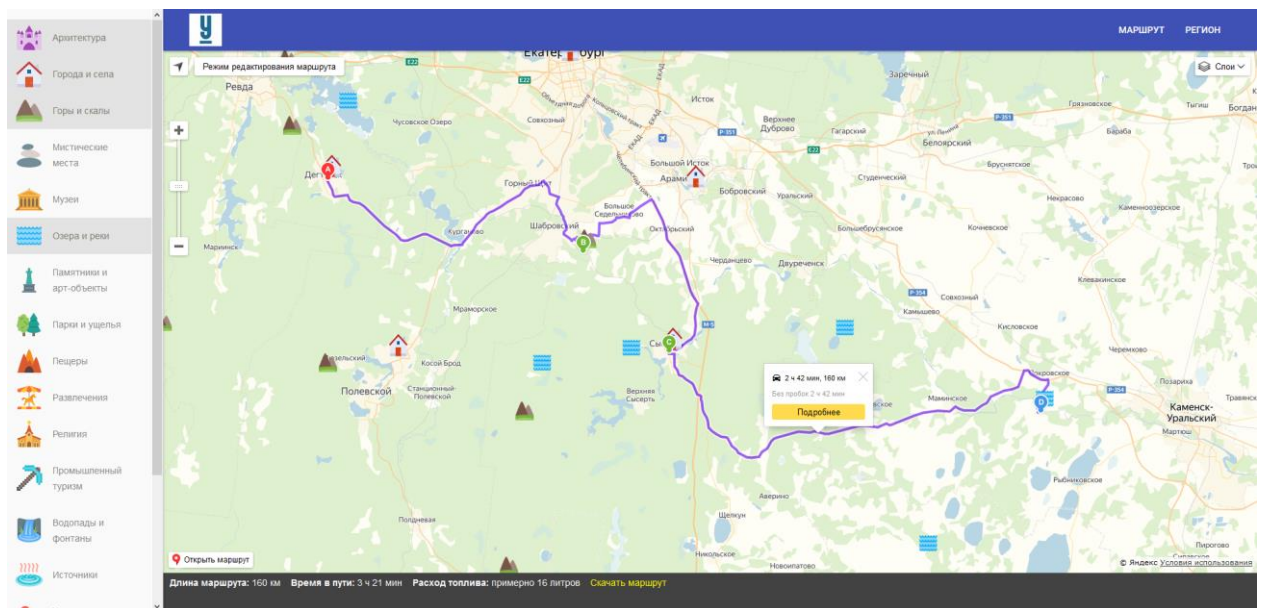


Рис 17 – Маршрут, созданный пользователем на карте

Разработка интерактивной карты достопримечательностей Урала закончена. После тестирования работы карты, она успешно введена в эксплуатацию на портале «Наш Урал».

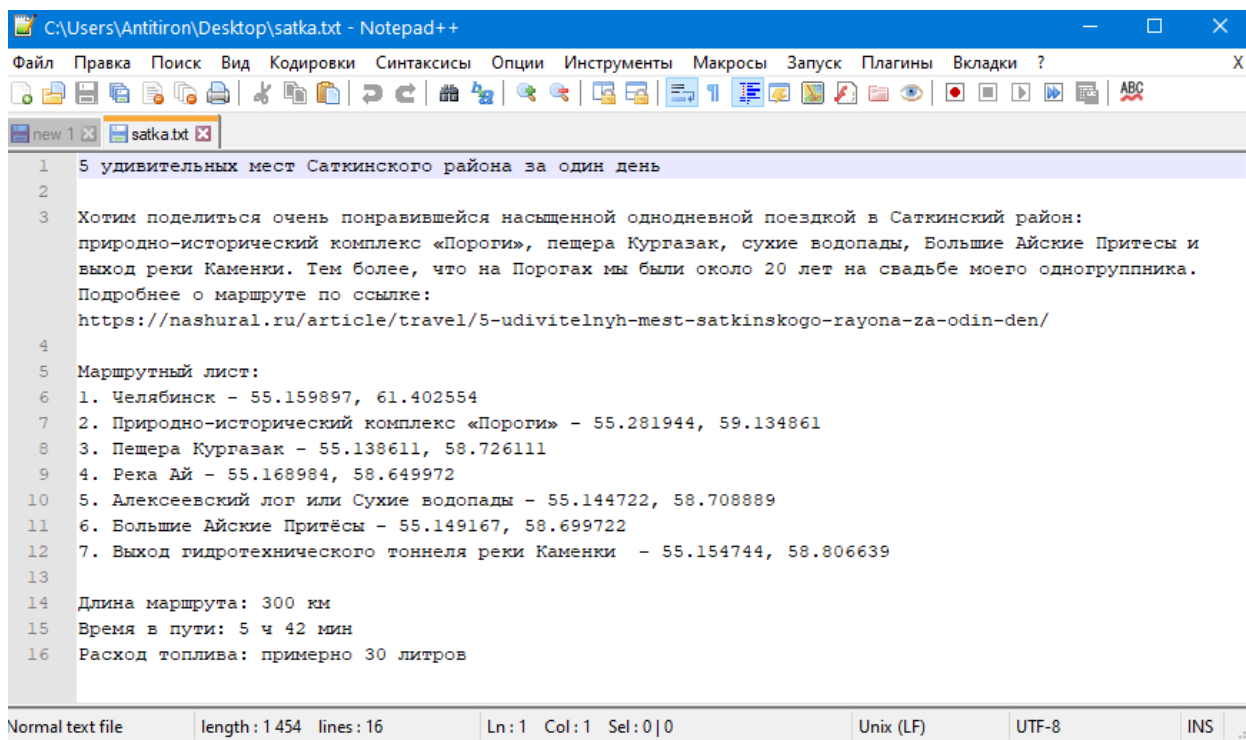


Рис 18 – Пример маршрутного листа

2.3 Апробация внедрения

После публикации интерактивной карты на портале «Наш Урал» было получено множество позитивных отзывов: пользователям действительно был необходим такой функционал, а сама интерактивная карта выполняет функции, которые возлагались на нее по техническому заданию.

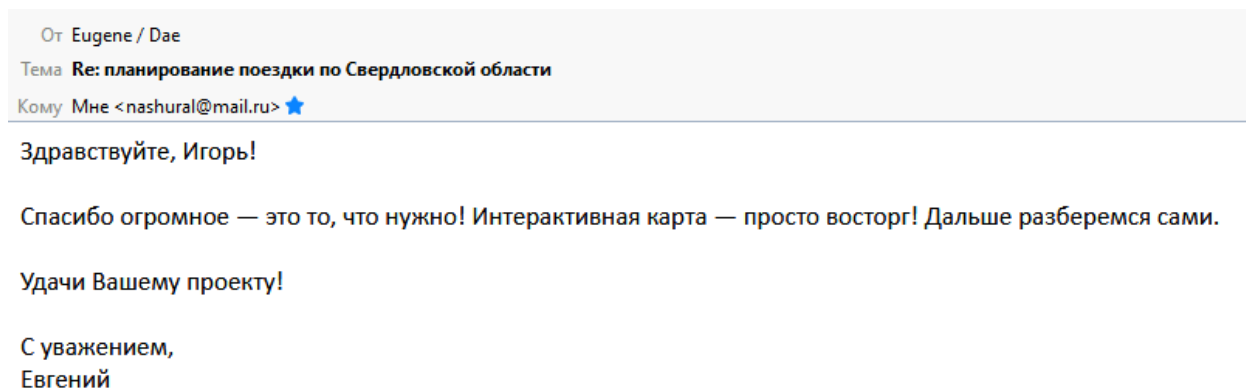


Рис 19 – Один из отзывов первого пользователя интерактивной карты

Пользователи с огромным интересом отнеслись к проекту и готовы предоставить свою помощь для улучшения проекта: исправление некоторых

неточностей, а также увеличение количества достопримечательностей на карте.

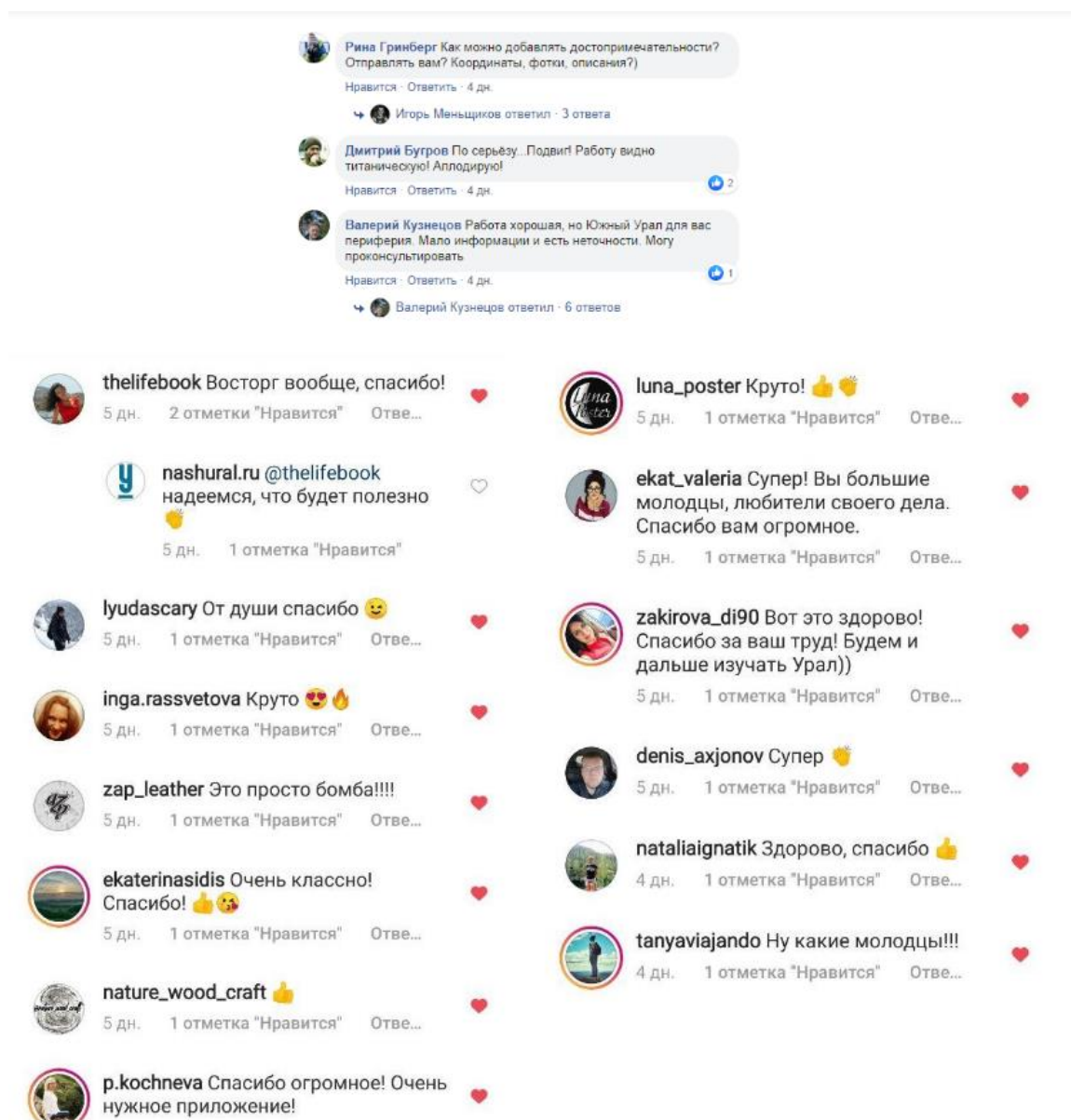


Рис 19 – Отзывы пользователей в соцсетях

В связи с этим встал вопрос о реализации возможности самостоятельного добавления пользователями объектов на карту, такой функционал планируется в дальнейшем при доработке проекта.

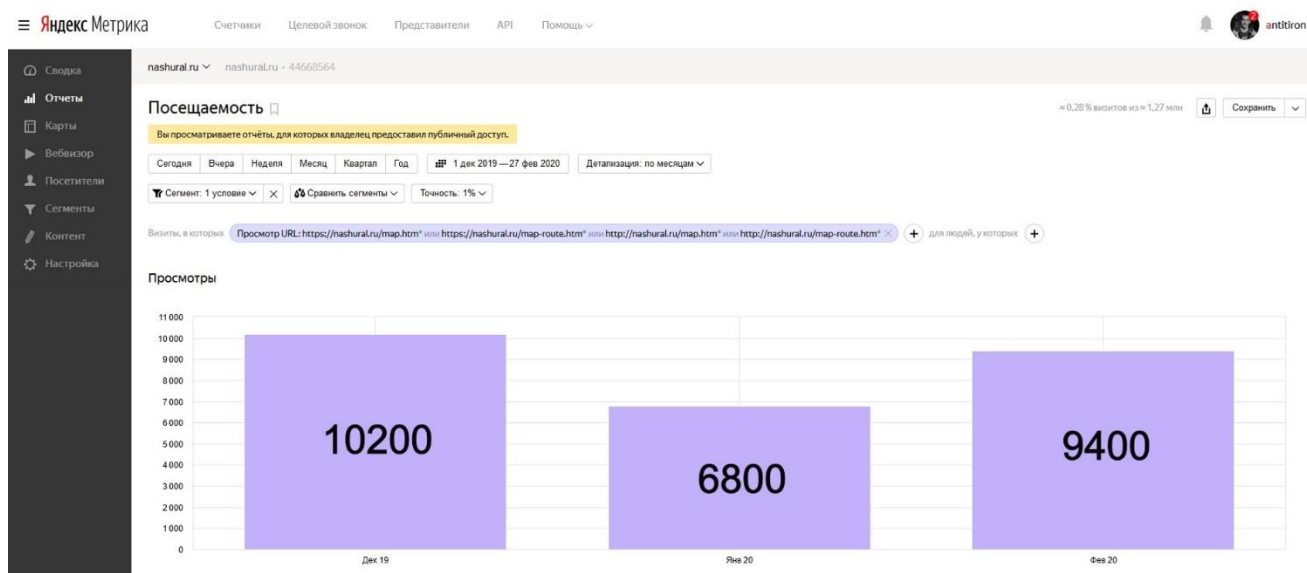


Рис 20 – Статистика просмотров карты

Отдельно стоит упомянуть, что проект легко масштабируем: можно создать такие интерактивные карты для любого другого региона России и мира. Все процессы автоматизированы и достаточно только подставлять необходимые базы данных с достопримечательностями и не только: карту можно использовать для любых других сфер деятельности, где есть необходимость в показе точек и построения маршрутов между ними. Главное условие на создание подобных карт – следовать лицензионному соглашению, потому что для многих сфер компания Яндекс представляет только платные возможности Яндекс.Карт.

Разработанная интерактивная карта достопримечательностей Урала внедрена на сайт в качестве самостоятельного раздела, успешно прошла апробацию, результаты которой представлены в Акте апробации в внедрения от 21 декабря 2019 года (см. приложение 1).

ЗАКЛЮЧЕНИЕ

Современные интерактивные технологии разнообразны и находятся в постоянном развитии. На их основе проектируются и разрабатывается множество самых разных сервисов и продуктов, которые призваны решить ту или иную потребность в жизни человека.

В ходе выполнения работы были рассмотрены различные интерактивные технологии, которые используются на современных веб-ресурсах. Были изучены популярные картографические сервисы и их веб-инструменты для создания собственных карт. По результатам этого анализа был выбран веб-инструмент для создания интерактивного решения: карты для построения туристических маршрутов по достопримечательностям Урала.

С помощью платформы Яндекс.Карты и их JavaScript API мы смогли реализовать удобную и наглядную навигацию по достопримечательностям, размещенных на портале «Наш Урал».

Мы также смогли создать функционал по построению маршрутов самими пользователями, предоставили им доступ к готовым маршрутам из списка самых популярных на портале «Наш Урал».

Разработанный продукт предполагает дальнейшее развитие: необходимо актуализировать базу достопримечательностей, исправить имеющиеся неточности, добавить возможность самостоятельного добавления пользователями новых достопримечательностей и маршрутов, реализовать поиск по достопримечательностям, провести работу по углубленной интеграции со статьями сайта «Наш Урал».

Несмотря на необходимость в доработке, уже сейчас интерактивная карта достопримечательностей Урала ежедневно используется посетителями портала.

Статистика позволяет говорить о том, что благодаря интерактивной карте пользователи стали быстрее находить известную им достопримечательность и с интересом рассматривают другие, находящиеся рядом. Формируют свои маршруты и с удовольствием рассматривают готовые

версии. Полученные от читателей положительные отзывы о работе с картой это подтверждают.

Таким образом, поставленные в работе задачи выполнены, цель достигнута.

СПИСОК ИСТОЧНИКОВ

1. Бейтс, М. CoffeeScript. Второе дыхание JavaScript / М. Бейтс ; пер. с англ. Киселёв А. – Москва : ДМК Пресс, 2016. – 310 с.
2. Брюс, Э. Философия Java / Э. Брюс ; пер. с англ. Матвеев Е. – Санкт-Петербург : Питер, 2019. – 1168 с.
3. Васильев, А. JavaScript в примерах и задачах / А. Васильев – Москва : Эксмо, 2017. – 720 с.
4. Веру, Л. Секреты CSS. Идеальные решения ежедневных задач / Л. Веру ; пер. с англ. Шикарева Е. – Санкт-Петербург : Питер, 2017. – 336 с.
5. Государев, И. Введение в веб-разработку на языке JavaScript / И. Государев – Санкт-Петербург : Лань, 2019. – 144 с.
6. Дакетт, Д. HTML и CSS. Разработка и дизайн веб-сайтов. JavaScript / Д. Дакетт ; пер. с англ. Райтман М. – Москва : Эксмо, 2017. – 480 с.
7. Дакетт, Д. Javascript и jQuery. Интерактивная веб-разработка / Д. Дакетт ; пер. с англ. Райтман М. – Москва : Эксмо, 2017. – 640 с.
8. Диков, А. Клиентские технологии веб-программирования: JavaScript и DOM / А. Диков – Санкт-Петербург : Лань, 2020. – 124 с.
9. Дронов, В. HTML, JavaScript, PHP и MySQL / В. Дронов, Н. Прохоренок – Санкт-Петербург : BHV, 2019. – 912 с.
10. Закас, Н. JavaScript. Оптимизация производительности / Н. Закас ; пер. с англ. Киселёв А. – Санкт-Петербург : Символ-Плюс, 2019. – 256 с.
11. Заяц, А. Проектирование и разработка WEB-приложений. Введение в frontend и backend разработку на JavaScript / А. Заяц, Н. Васильев – Санкт-Петербург : Лань, 2019. – 120 с.
12. Кириченко, А. Динамические сайты на HTML, CSS, JavaScript и Bootstrap / А. Кириченко, Е. Дубовик – Санкт-Петербург : Наука и Техника, 2018. – 272 с.

13. Кириченко, А. JavaScript для FrontEnd-разработчиков. Написание. Тестирование. Развертывание / А. Кириченко – Санкт-Петербург : Наука и Техника, 2020. – 320 с.
14. Крокфорд, Д. Как устроен JavaScript / Д. Крокфорд ; пер. с англ. Вильчинский Н. – Санкт-Петербург : Питер, 2019. – 304 с.
15. Крокфорд, Д. JavaScript. Сильные стороны : монография / Д. Крокфорд ; пер. с англ. Лузган А. – Санкт-Петербург : Питер, 2017. – 176 с.
16. Макдональд, М. Веб-разработка. Исчерпывающее руководство : монография / М. Макдональд ; пер. с англ. Черников С. – Санкт-Петербург : Питер, 2017. – 640 с.
17. Макконнелл, С. Совершенный код : монография / С. Макконнелл ; пер. с англ. Вшивцев В. – Москва : Русская Редакция, 2019. – 896 с.
18. Макфарланд, Д. JavaScript и jQuery. Исчерпывающее руководство / Д. Макфарланд ; пер. с англ. Райтман М. – Москва : Эксмо, 2017. – 880 с.
19. Морето, С. Bootstrap в примерах / С. Морето ; пер. с англ. Рагимов Р. – Москва : ДМК Пресс, 2017. – 314 с.
20. Роббинс, Д. HTML5, CSS3 и JavaScript. Исчерпывающее руководство: монография / Д. Роббинс ; пер. с англ. Райтман М. – Москва : Эксмо, 2017. – 528 с.
21. Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 : монография / Р. Никсон ; пер. с англ. Вильчинский Н. – Санкт-Петербург : Питер, 2017. – 768 с.
22. Петин, В. API Яндекс, Google и других популярных веб-сервисов / В. Петин – Санкт-Петербург : BHV, 2017. – 480 с.
23. Прасти, Н. Введение в ECMAScript 6 : монография / Н. Прасти ; пер. с англ. Рагимов Р. – Москва : ДМК Пресс, 2017. – 176 с.
24. Резиг, Д. JavaScript для профессионалов / Д. Резиг, Р. Фергюсон, Д. Пакстон ; пер. с англ. Берштейн И. – Москва : Вильямс, 2017. – 240 с.
25. Резиг, Д. Секреты JavaScript ниндзя : монография / Д. Резиг, Б. Бибо, И. Марас ; пер. с англ. Берштейн И. – Москва : Вильямс, 2017. – 544 с.

26. Симпсон, К. ES6 и не только : монография / К. Симпсон ; пер. с англ. Рузмайкина И. – Санкт-Петербург : Питер, 2017. – 336 с.
27. Фаулер, М. Рефакторинг кода на JavaScript. Улучшение проекта существующего кода / М. Фаулер ; пер. с англ. Красиков И. – Москва : Вильямс, 2019. – 466 с.
28. Фельке-Моррис, Т. Большая книга веб-дизайна / Т. Фельке-Моррис ; пер. с англ. Райтман М. – Москва : Эксмо, 2017. – 608 с.
29. Финков, М. Создание сайтов своими руками на BOOTSTRAP : монография / М. Финков, А. Евдокимов – Санкт-Петербург : Наука и Техника, 2017. – 240 с.
30. Флэнаган, Д. JavaScript. Подробное руководство / Д. Флэнаган ; пер. с англ. Киселева А. – Санкт-Петербург : Символ-Плюс, 2017. – 1080 с.
31. Флэнаган, Д. JavaScript. Карманный справочник / Д. Флэнаган ; пер. с англ. Сысолюк А. – Москва : Вильямс, 2018. – 320 с.
32. Фримен, Э. Изучаем программирование на JavaScript / Э. Фримен, Э. Робсон ; пер. с англ. Матвеев Е. – Санкт-Петербург : Питер, 2017. – 640 с.
33. Хавербеке, М. Выразительный JavaScript. Современное веб-программирование / М. Хавербеке ; пер. с англ. Сандицкая Е. – Санкт-Петербург : Питер, 2019. – 480 с.
34. Хрусталеv, А. HTML5 + CSS3. Основы современного WEB-дизайна: монография / А. Хрусталеv, А. Кириченко – Санкт-Петербург : Наука и Техника, 2018. – 352 с.
35. Шилдт, Г. Java. Полное руководство / Г. Шилдт ; пер. с англ. Бернштейн И. – Санкт-Петербург : Вильямс, 2018. – 1488 с.
36. Эккель, Б. Философия Java / Б. Эккель ; пер. с англ. Матвеев Е. – Санкт-Петербург : Питер, 2019. – 1168 с.
37. Эспозито, Д. Разработка современных веб-приложений. Анализ предметных областей и технологий / Д. Эспозито ; пер. с англ. Ключина Д. – Санкт-Петербург : Вильямс, 2017. – 464 с.

38. API v0.6 OpenStreetMap Wiki : [сайт] – Лондон, 2019 – . – URL: http://wiki.openstreetmap.org/wiki/API_v0.6 (дата обращения: 01.12.2019)
39. Bootstrap Documentation : [сайт] – Вашингтон, 2019 – . – URL: <https://getbootstrap.com/> (дата обращения: 08.12.2019)
40. Google Maps JavaScript API : [сайт] – Вашингтон, 2019 – . – URL: <https://developers.google.com/maps/documentation/javascript/> (дата обращения: 02.12.2019)
41. JavaScript API Яндекс.Карт : [сайт] – Москва, 2019 – . – URL: <https://tech.yandex.ru/maps/jsapi/> (дата обращения: 03.12.2019)
42. PhoneGap Documentation : [сайт] – Вашингтон, 2019 – . – URL: <http://docs.phonegap.com/> (дата обращения: 07.12.2019)
43. YMapsML – Технологии Яндекса : [сайт]. – Москва, 2019 – . – URL: <https://tech.yandex.ru/maps/ymapsml/> (дата обращения: 04.12.2019)

ПРИЛОЖЕНИЯ

Приложение 1

АКТ АПРОБАЦИИ И ВНЕДРЕНИЯ

интерактивной карты для построения туристических маршрутов по
достопримечательностям Урала

Составлен «21» декабря 20 19 года.

В ООО «Real Russian Expedition» (г. Екатеринбург Свердловской области) в рамках выпускной квалификационной работы «Интерактивная карта для построения туристических маршрутов по достопримечательностям Урала» была проведена апробация и внедрение карты на портал «Наш Урал» (<https://nashural.ru>).

В процессе апробации были выполнены следующие работы:

1. Проведен анализ существующих интерактивных веб-технологий;
2. Создана база данных достопримечательностей Урала;
3. Спроектирована и разработана интерактивная карта для построения туристических маршрутов по достопримечательностям Урала;
4. Протестирована и опубликована карта на портале «Наш Урал»;
5. Подготовлена инструкция по работе с картой для пользователей.

Заключение по результатам апробации и внедрения интерактивной карты для построения туристических маршрутов по достопримечательностям Урала:

1. Работы по разработке интерактивной карты и плану апробации проведены в полном объеме.
2. Интерактивная карта удобна в использовании и содержит все необходимые функции для построения туристических маршрутов по достопримечательностям Урала.

Генеральный директор «Real Russian Expedition»



Чечулин А.В.